中国教育电子公司

中华学习机
CEC—ⅠA
用户使用手册

清华大学出版社



中华学习机CEC-IA
用户使用手册

中国教育电子公司

清 华 大 学 出 版 社

内 容 简 介

中华学习机CEC-IA是中华学习机CEC-I的增强型。该机扩充了许多功能，它的软件开发环境是极好的。本书是用户使用手册，内容包括：基本系统的操作使用，汉字系统的操作使用，CEC-BASIC语言的使用方法，监控与小汇编的使用方法，磁盘操作系统的使用，打印机的安装使用等。

中华学习机配上家用电视或监视器，再接上录音机或软盘驱动器就可以构成一台功能较强，具有汉字支持的微机系统。它是青少年理想的工作、学习、娱乐工具。本书可供青少年、中小学教师及家长阅读，也可供计算机应用人员参考。

中华学习机CEC-IA用户使用手册

中国教育电子公司



清华大学出版社出版
北京　清华园
北京朝阳区科普印刷厂印刷

开本：787×1092 1/32　印张：10 5/8　字数：240（千）
1990年2月第1版　1990年2月第1次印刷
印数：0001～2000
ISBN 7-302-00661-X/TP·227
定价：9.50元

前　言

中华学习机CEC-ⅠA是中华学习机CEC-Ⅰ的增强型。该机种兼容AppleⅡe，具有80列显示功能，即每行可显示80个字符。由于扩充了双驱动器、打印机及串行通信接口电路，因此可以直接连接两个驱动器，一个打印机及进行串行通信。存储器扩充到256K（ROM128K，RAM128K），键盘增至87个键。另外，还增加了三声道的音响合成器。汉字系统增加了五笔字型和笔型码输入方法。具有多字符组、造字、放大等功能，对开发教学软件具有极好的开发环境。

如果你仔细看一下本书目录就可以发现许多新增加的功能和语句，当你熟悉了CEC-ⅠA的使用方法时就会相信它的软件开发环境是同档机中最优秀的。

欢迎你使用中华学习机CEC-ⅠA，愿它成为你的好伙伴。

编　者　潘孝梅　卓小越　周　童
武光鼎　卜文璿

目　录

第一章　CEC-IA 中华学习机简介

1.1 系统的组成

一套CEC-ⅠA中华学习机是由主机盒(包括键盘），显示器以及外存储器组成。主机盒配上显示器及盒式录音机便构成基本系统。图1-1就是带录音机的基本系统。

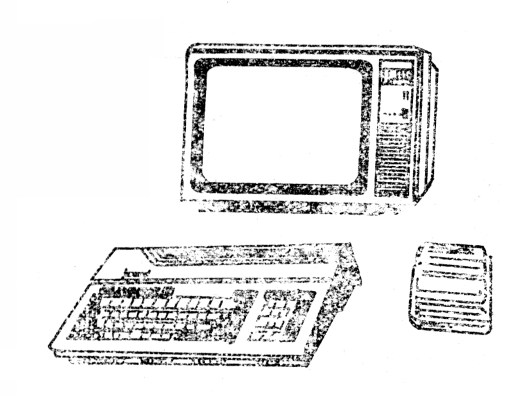


图 1-1　带录音机基本系统

如果用软盘驱动器代替盒式录音机作为外存就可以运行磁盘操作系统，构成一台功能较强的而且有汉字支持的微机系统(如图1-2）。

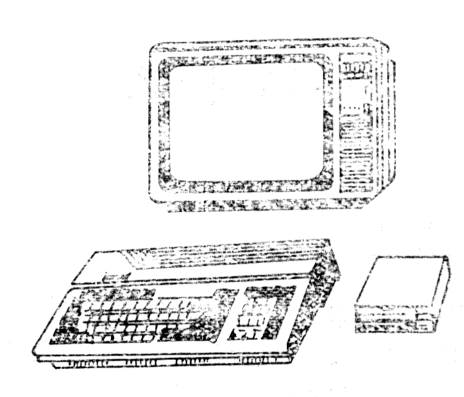


图 1-2 带软盘驱动器的微机系统

1.1.1 键盘

键盘是计算机的最基本的输入设备，是人-机对话的主要工具之一。通过键盘可以打入各种命令和数据，控制计算机的工作。CEC-ⅠA的键盘安装在主机盒内，组成一体化的结构，共有87键。包括了标准打字机键盘的各种键，并按打字机的键位排列，此外，还有特殊用途的功能键，整个键盘的排列如图1-3所示。

从图中可知，键盘的右侧为键盘小区，其中以数字键及常用的功能键为主，这种排列方式对财会系统及有些汉字输入方法的使用带来了方便。

1.1.2 显示器

显示器是计算机的最基本输出设备。通过显示器可以查看当前计算机的工作状态和执行结果。CEC-ⅠA中华学习

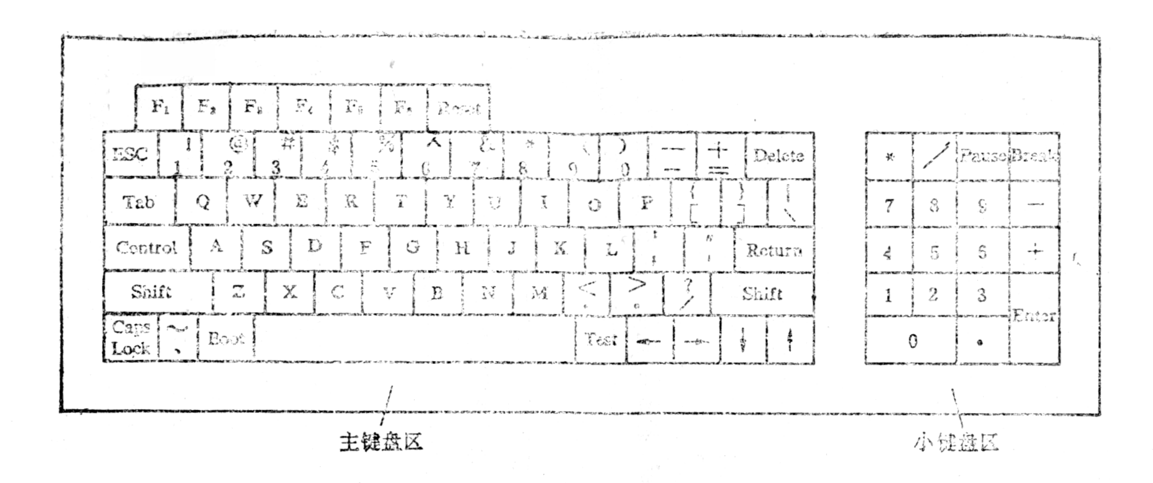


图 1-3 CEC-IA 键盘排列图

机采用以复合视频信号方式工作的单色或NTSC/PAL两种制式的彩色监视器，也可连接R.G.B直接驱动方式工作的彩色监视器。此外，通过射频调制器（外部选件）可以连接黑白或NTSC/PAL-D两种制式的彩色电视机作为系统的显示器。

1.1.3 盒式录音机

盒式录音机要求有录放功能，并带有EAR和MIC 3.5mm插座，以单声道的录音机为好。由于家用的盒式录音机性能差别甚大，建议使用工厂推荐的配套盒式录音机。

此外，CEC-ⅠA具有控制录音机启停的功能，因此，具有‘Remote’控制插孔的录音机更为理想。

1.1.4 软盘驱动器

CEC-ⅠA可以连接两台5.25英寸单面单密度软盘驱动器，每台容量143K字节。也可以连接一台专用的5.25英寸双面单密度软盘驱动器，其容量为286K字节。

1.1.5 主机盒

CEC-ⅠA的电路部分做在盒内主机板上，包括中央处理器6502，存储器电路，以及输入输出接口电路。系统的电源也安装在主机盒内。

中央处理器6502为八位微处理器，时钟频率为1MHz，寻址范围64K字节。

存储器由RAM和ROM两部分组成。RAM分为主存储器和辅存储器，容量各为64K字节，由4片64K×4的动态存

储器芯片组成，存取时间为120ns（或150ns）。ROM容量为128K字节，由2片EPROM 27512组成。其中一片固化有监控程序、CEC-BASIC、汉字管理程序以及输入输出设备的驱动程序。另一片固化了汉字输入转换码表，亦可固化用户应用程序。

主板上使用了7片专用集成电路，即存储器管理部件MMU、AUMMU，键盘编码电路KBC，软盘控制器FDC，彩色编码电路CEU，输入输出部件IOU，以及可编程阵列逻辑电路。

主机的左侧有一个与AppleⅡ兼容的50线输入输出插座，用于扩充功能。输入输出接口电路和电源在下面两节说明。

1.1.6 输入输出接口

CEC-ⅠA的输入输出接口电路包括键盘输入接口，显示器输出接口，盒式录音机输入输出接口，软盘驱动器接口，打印机接口，RS-232C串行接口，音响发生器接口以及游戏控制杆接口。

一、键盘输入接口

此接口电路将键盘产生的位置码变换成按键所对应的ASCII码输入到主机内。在主机板上有一个26芯插头，由扁平电缆与键盘相连。

二、显示器接口

显示器接口的作用是将计算机要输出的字符或图形数据变为显示器可接收的视频信号送到显示器。CEC-ⅠA显示器接口电路产生NTSC/PAL制式彩色复合视频信号，送到

标有“MONITOR”插座上，此信号经外部的调制器输出NTSC/PAL制式的射频信号，可以连接两种制式的彩色电视机，接口电路还输出直接驱动的R。G。B三基色信号，送到标有“R。G。B”的D9插座上。

三、盒式录音机接口

录音机接口电路将需要录入的程序或数据转换成录入磁带的写入信号记录到磁带上，同时也能把磁带上读出的信号整形放大转换成数据，存放到计算机内存中。接口电路的输出与录音机的MIC插孔连接，接口电路的输入与录音机的EAR端相连。此外，还有控制启停的信号与录音机的“Remote”端相连接。

四、软盘驱动器接口

软盘驱动器接口可连接两台5.25英寸单面单密度软盘驱动器。通过换接，也可在1号软盘驱动器插头上连接一台专用的5.25英寸双面单密度的软盘驱动器。

五、扬声器及音响发生器输出接口

主机盒内有一只0.25W8Ω的扬声器，由扬声器接口电路驱动，通过运行程序可发出所需要的乐曲声。

此外，在主机板上有一片音响发生器电路，可以发出悦耳的电子乐器的音色和效果。主机盒的后侧有一个“Sound”插孔，经转换可以输出机内扬声器或音响发生器电路产生的音频信号。

六、打印机接口

打印机接口电路产生Centronic标准的九针打印机所需要的信号,可以连接如CP-80，MX-80，RX-80，FX-80等九针打印机。

七、串行接口

主机板上具有标准RS-232C串行接口电路，主机盒后侧面提供一个RS-232C的D25插头。

八、游戏控制杆接口

此接口电路提供三个TTL电平的开关量输入和四个模拟量输入端口，由D9九芯插座将游戏控制杆信号输入到接口电路，该接口电路与AppleⅡ微机兼容。

九、扩充槽口

CEC-ⅠA上保留1个与AppleⅡ兼容的50芯插座，定义为5号插座。同时也将IOSEL7，DEVSEL7信号引到此插座的空腿上，以备使用。

1.1.7 电源

主机盒内装有一个开关电源，输出+5V，+12V，-5V，-12V四种直流电源，最大交流功耗为30W，各路电压的额定负载电流为：

+5V	2.0A
+12V	1.0A
-5V	0.1A
-12V	0.1A

1.2 主要技术指标

一、主机

1. 中央处理器：6502，8位微处理器。
2. 内存容量：

   RAM：128K字节。

ROM：128K字节，其中64K字节固化监控程序，CEC-BASIC语言及汉字管理程序，另外64K字节可以扩充汉字输入方法或其它用户程序。

3．采用7片专用集成电路：分别为MMU、AUMMU存储器管理部件，IOU输入/输出管理部件，CEU彩色编码电路，FDC软盘控制器，KBC键盘编码电路，以及时序电路。

4．汉字功能部件：此部件已做在主板上。本机以拼音、区位、王码及笔形码作为基本输入法，并已固化在主板ROM中。采用全点阵的国标一、二级汉字字库，包括6763个汉字及外文字母等。

5．显示器接口：一个是复合视频信号输出，可连接NTSC/PAL制式彩色或单色监视器，复合视频信号经过带声音输入的射频调制器可连接两种制式的彩色或黑白电视机。另一个是输出R．G．B直接驱动信号，可连接该驱动方式的彩色监视器。

6．软盘驱动器接口：可连接两个5.25英寸软盘驱动器，或连接一台专用的5.25英寸双面单密度软盘驱动器。

7．盒式录音机接口：可与一般家用或专用录音机连接，接口电路输出25mV电压信号作为录入信号，要求录音机的读出信号≥1V（峰-峰值），电路的输出阻抗为100Ω，输入阻抗为12kΩ。

8．键盘与键盘接口：为标准打字机键盘，共87键，包括字母，数字，及一些特殊控制功能键，采用弹簧键体。

9．RS-232C接口，该电路已做在主板上。

10．并行打印机接口：可连接具有Centronic标准接口的打印机。

11．游戏控制杆接口：提供三个TTL电平的开关量输入，四个模拟量输入，可连接游戏控制杆，另外在主板上有一个16线IC插座，提供游戏杆控制信号以及四个软开关信号。

12．音响发生器电路：该电路提供三个独立声道，经合成产生不同乐器的音色和效果。

13．扩充槽口：提供1个与AppleⅡ兼容的50线I/O插座。

14．主机盒内装有开关电源，最大交流功耗为30W。

二、显示器

可以使用具有复合视频信号输入的NTSC/PAL制式彩色或黑白监视器，亦可使用具有R．G．B直接驱动方式输入的彩色监视器。

屏幕显示功能：

1．西文方式

字符构成：5×7点阵

每帧字符：40字符×24行，80字符×24行两种。

显示方式：正常、反相、闪烁（仅适用于每行40字符）

2．中文方式

字符构成：16×16点阵（汉字）

8×16点阵（ASCII）

每帧字符：每帧为12行（也可设置成11行）

每行字符数：

汉字为17和34字符两种。ASCII字符为34和68字符两种。其中一行可设置成状态行。

3．图形方式

低分辨率彩色图形显示，分辨率为40H×48V16种颜色。

高分辨率彩色图形显示，分辨率为280H×192V6种颜色。

倍高分辨率彩色图形显示，分辨率为560H×192V16种颜色。

三、软盘驱动器

采用5.25英寸单面单密度软盘驱动器，也可采用专用的5.25英寸双面单密度软盘驱动器。

四、盒式磁带机

任选一般盒式录音机或专用录音机。

五、打印机

可直接与具有Centronic接口标准的打印机连接。

六、软件

系统软件：监控程序，DOS3.3，CEC-BASIC以及汉字管理软件。

应用软件：多种教学辅助软件和游戏程序以及和Apple Ⅱ兼容的应用软件。

第二章　基本系统的操作使用

2.1　系统的安装

2.1.1　主机

主机是一个结构紧凑的长方形机盒。机盒上面前半部为键盘，右上角并列两只红色指示灯。左侧为电源指示灯，右侧为大写字符锁定键指示灯。开机后，电源指示灯即亮，大写字符锁定指示灯也处于亮的位置。此灯亮表示输入的英文字符为大写字符，不亮表示输入的英文字符为小写字符，通过按下CapLock键可以改变其状态。机盒左侧为扩展槽口，右侧为电源开关，后侧为各种接口。

一、主机接口

主机的接口参照图2-1。

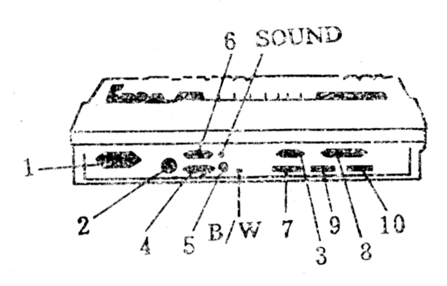


图　2-1

1. 电源插座（～220V）
2. 录音机插座（TAPE RECORDER）

3. 游戏杆插座（JOYSTICK）

4. 音频、视频外接口（TV VIDEO）

5. 监视器插座（MONITOR）

6. 直接驱动监视器接口（R。G。B）

7. 打印机接口插座（PRINTER）

8. 串行接口（RS-232C）

9. 驱动器A接口（DRIVE 1）

10. 驱动器B接口（DRIVE 2）

位于机盒左侧，有一50芯扩展槽口，如图2-2 所 示。

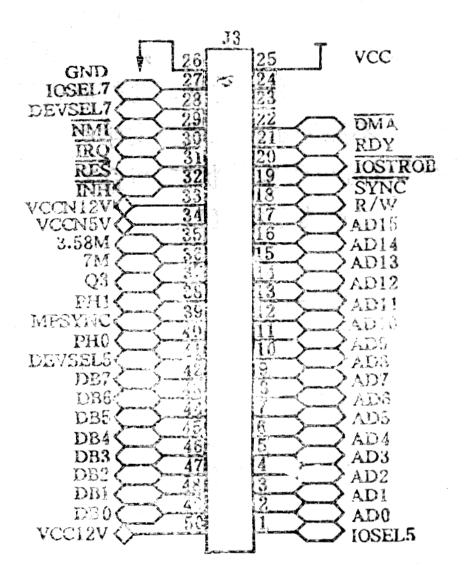


图 2-2

二、内部开关

图2-3为机盒内主机板上开关位置图。

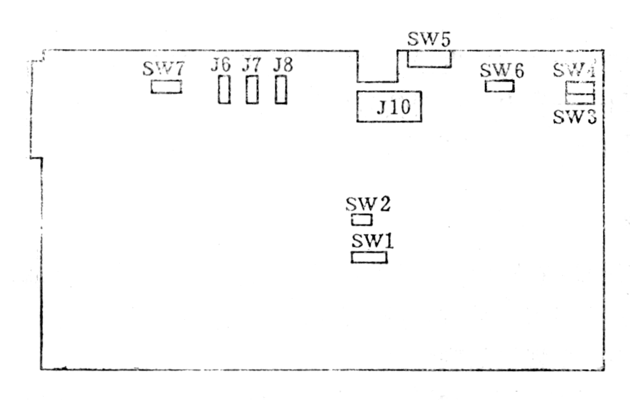


图 2-3

SW1——U15 ROM选用27512时，将插针的1，2短接，选用27256时将2，3短接。

SW2——使用监视器或电视机为NTSC制式时，将此插针短接，选用PAL制式时，将此断开。

SW3
SW4——使用有“REMOTE”控制的录音机时，将插针的1，2短接，或将2，3短接，具体随录音机而定。

SW5——黑白/彩色转换开关（可从外部使用）。

SW6——音频、视频外接口中音频信号的选择。将1,2短接时，向外输出机内喇叭信号，将2，3短接时，向外输出音响发生器电路MSC产生的音频信号。

SW7——驱动器选择开关。将1，2短路时可接A、B双驱动器，将2，3短路时，B驱动器插座（DRIVE 2）无效，A驱动器插座除可使用普通驱动器之外，还可以使用一种特殊的驱动器，此驱动器可使用磁盘的正反面，软件选中A驱动器时使用磁盘的正面，选中B驱动器时，使用磁盘的反面。

J6——机内喇叭接口，此两针之间可接一只8Ω扬声器。

J7——音响发生器电路MSC输出外接口。

J8——机内扬声器信号外接口。

J10——输入输出接口。

2.1.2 显示器的联接

CEC-ⅠA中华学习机可以选用以下任一种显示器：电视机、监视器、带有音频的监视器。

一、电视机的联接

1．电视机是通过RF与主机相联的，RF的一端插在主机音频、视频外接口（TV VIDEO）上（图2-1），另一端通过视频线接入电视机天线外接插座，输入阻抗为75Ω。连接方法参见图2-4。

2．将电视机与主机均打开，同时按下主机键盘的Ctr-Reset-Test键，进入自检（后抬起Test键），连续敲击两次空格键，屏幕显示彩条状态。

3．按RF说明将电视调到相应频道，并仔细调好。

4．使用NTSC制式彩色电视机时，除以上步骤外，还

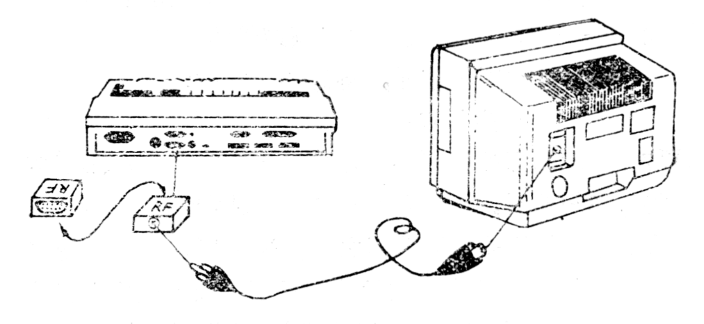


图 2-4

应先将机内SW2短接(图2-3)，出厂时主机按PAL制连接。

二、监视器的联接

1. 普通监视器通过视频线与主机联接，视频线一端联主机的监视器接口（MONITOR），另一端接监视器信号输入端。接线方法参阅图2-5。

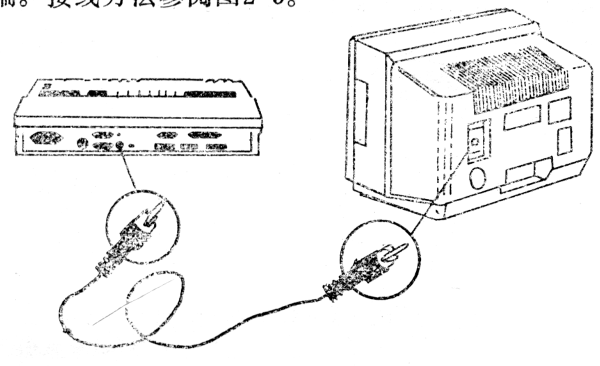


图 2-5

2. 带有音频输入端的监视器，以同样方法联接视频线，并且还要联接音频，音频可由电缆接主机内部的J7或J8（图2-3上端为信号，下端为地）也可由音频、视频外接口(图2-1)的第五针接出音频。电缆另一端接入监视器音频输入口。

3. NTSC制监视器，使用时还须将机内SW2短接（图2-3）。

2.1.3 录音机的联接

录音机通过随机专用电缆与主机联接，联接线一端为单一的五芯插头，插入主机盒的录音机插座内（TAPE RECORDER）（图2-1），另一端的红头（ϕ3.5）插入录音机的耳机插孔（EAR)，黑头（ϕ3.5）插入录音机外接MIC

插孔，带有遥控的录音机将细的插头（φ2.5）插入录音机的REMOTE孔内。并根据录音机将主机内的开关SW3、SW4设置好。出厂时设置为2、3短接，一般可直接使用，若不能使用时，可将SW3、SW4的2、3同时断开而将1、2同时短接。录音机电缆连接方法参阅图2-6。

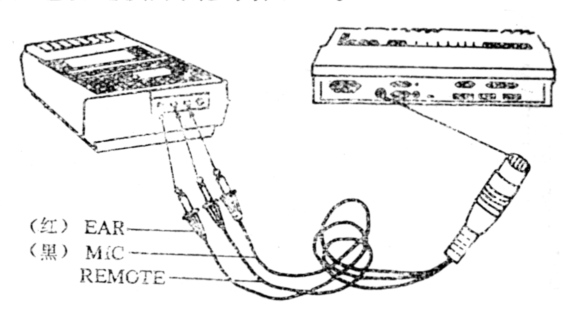


图 2-6

2.1.4 软磁盘驱动器的联接

CEC-ⅠA可接两台单面140KB5.25英寸软磁盘驱动器，也可使用特殊的双面驱动器。联接普通驱动器时，直接将驱动器电缆插入接口定位槽内。联接双面驱动器时，将驱动器插在A槽（DRIVE1）（图2-1）并将机内开关SW7的2、3短接，1、2断开（出厂时将2、3断开，1、2短接）。驱动器的联接参照图2-7。

2.1.5 打印机的联接

打印机电缆的一端为双列十针插座，用于联接主机打印机接口，插于定位槽内，另一端接入打印机。连接方法参照图7-5，图7-6。

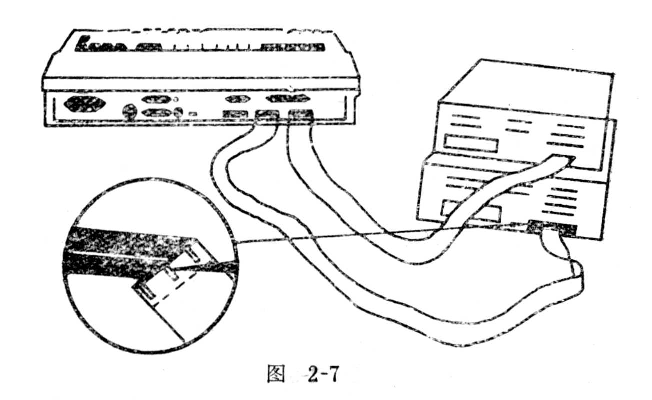


图 2-7

2.1.6 串行接口

主机提供标准的RS-232C串行接口，接口为D型25芯插针，提供联线如下：

D25针号	信号名称
1，7	信号地GND
2	发送数据TxD
3	接收数据RxD
4	请求发送RTS
5	清除发送（发送准备好）CTS
6	数据设备准备好DSR
20	数据终端准备好DTR
8	信号检测DCD

通信电缆的联接参见图2-8。

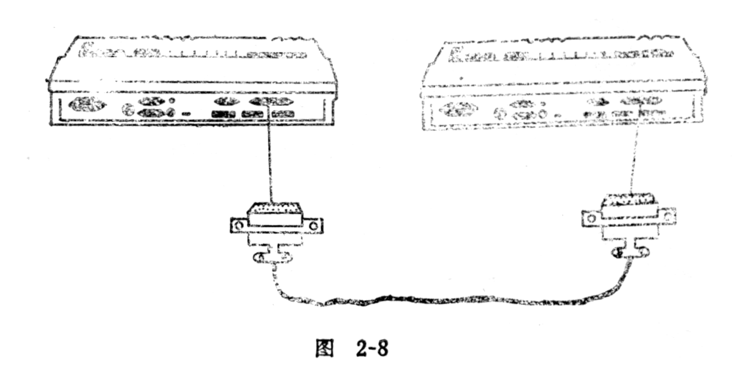


图 2-8

2.1.7 游戏杆的联接

游戏杆通过D9型插座与主机联接，联接时，将游戏杆插入主机盒游戏杆插座内（JOYSTICK）。插座引脚分配如图2-9。

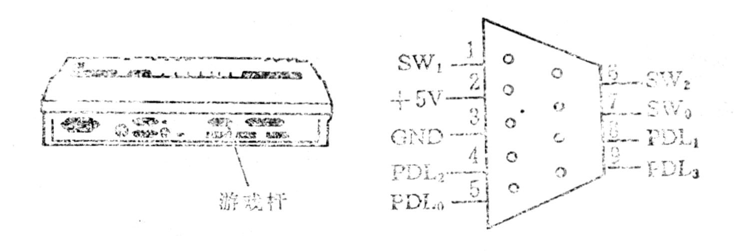


图 2-9

2.1.8 扩充卡的插接

CEC-ⅠA可插接各种CEC-Ⅰ型机及APPLE-Ⅱ机所配外接扩充卡，插入时元件面向上。50芯插座的引脚定义如

图2-2。

*注意：插拔各种接口及各种扩充卡时，必须将主机及外设（打印机等）切断电源，否则容易引起机内接口电路的损坏。

2.2 机器的启动

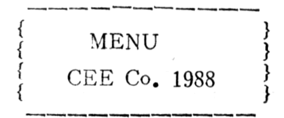
2.2.1 基本系统的启动

基本系统包括主机、监视器、录音机等。将以上系统联接好，打开主机电源开关，屏幕上出现：

CHINA EDUCATION COMPUTER

VERSION 2. 0

经过数秒钟后出现一菜单：



0. CEC-BASIC
1. CHINESE - BASIC
2. 80COL-BASIC
3. MONITOR

PLEASE GIVE YOUR CHOICE IN 0—3

并等待输入。分别键入数字后，系统将进入指定状态，此时文件的存取可通过录音机进行。

2.2.2 联接驱动器的系统启动

主机可使用两台驱动器，若只用单台驱动器时，应放在A号槽（DRIVE1）。将DOS盘插入驱动器。开机后系统调用磁盘，数秒钟后屏幕出现：

```
DOS VERSION 3. 3                08/25/80
APPLE Ⅱ PLUS OR ROMCARD  SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CARD)
]
```

此时即进入DOS状态，可通过磁盘存取文件。

2.2.3 键盘冷启动

为方便用户，减少开关电源对主机系统的冲击，特增设键盘冷启动功能，使用时将主机Ctrl-Reset-Boot 3键同时按下，并最后抬起Boot键，系统即进入启动状态。

2.3 键盘操作

键盘是计算机的基本输入设备。人通过键盘把命令输入计算机，计算机通过键盘接收到人的命令并执行。因此键盘操作是学会使用计算机的基本内容。中华学习机的键盘提供了丰富的操作功能，它为用户的操作和使用带来极大方便。

2.3.1 键盘的分区和字符键

键盘上共有87个键。这些键分成两个区，即主键盘区69个键和小键盘区18个键。参见图2-10。主键盘区用于计算机的基本操作。小键盘区主要是数字键，在银行和财务管理等部门使用较多。

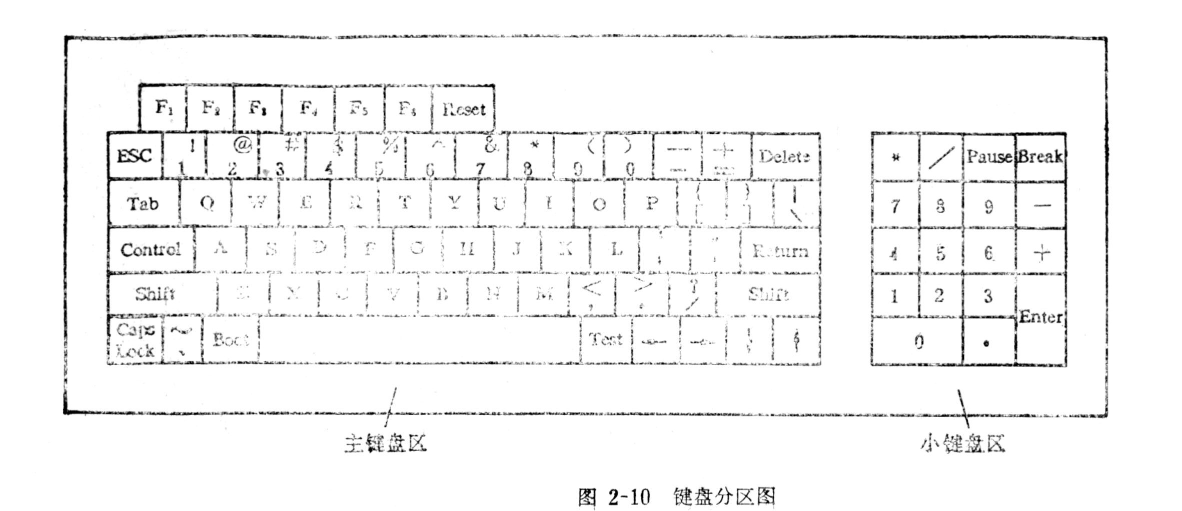

F1
F2
F3
F4
F5
F6
Reset
ESC
Tab
Control
Shift
Caps Lock
Boot
Test
Delete
Return
Shift
Pause
Break
Enter
主键盘区
小键盘区

图 2-10 键盘分区图

主键盘区的键分为字符键和其他功能键。其中0，1,2，…，9为数字键，A，B，C,…,Z为字母键，等号“=”，减号“-”，句号“。”等为符号键。我们把数字键、字母键和符号键统称为字符键。这些字符键是我们经常使用的，操作也很方便。例如按下A键，屏幕上就会出现一个“A”。另外，键盘最下面的那个长键称为空格键，也属于字符键。按一下空格键，屏幕上的光标就向右移一格，字符显示出现一个空格，即这个位置什么也不显示。

除字符键外，还有一些特殊键，它们的作用介绍如下。

2.3.2 Shift键

这个键共有两个，分别安排在键盘的左侧和右侧，这是为了操作起来方便。这个键的功能是当按此键和另一键时，产生上档键码。例如，在标有数字“8”的键上还标有星号“*”。如果要把“*”显示到屏幕上就必须将Shift与“8”这两个键同时按下，或者先按下Shift键不动，然后再按一下“8”。同样，凡是键上有上下两个字符的，要想使用上面的那个字符就要借助于Shift键。

2.3.3 Caps Lock键

Caps Lock键是英文Capitals Lock的缩写，意思是英文“大写字母锁定”。开机时，Caps Lock键的指示灯亮着，表示此键处于大写字母锁定的位置。此时，键盘上输入的英文字母都是大写字母。按一下Caps Lock键，Caps Lock键的指示灯灭，表示此时没有处于大写锁定。此时键盘上的英文字母为小写字母。若再按一下Caps Lock键，英文字

母又回到大写状态。在小写状态下若要临时输入一、二个大写字母可以按下Shift键不动，然后按你要输入的大写字母。当Shift键抬起后，字母又回到小写状态。

2.3.4 Ctrl键

Ctrl是英文Control的缩写。是“控制”的意思。Ctrl键总是与另外一个键同时使用，即按住Ctrl键不放，再按另一键，使得同一键具有另外的功能。例如，当“G”键与Ctrl键同时使用时，我们用Ctrl-G来表示，这时计算机会发出“嘟”的一声。这就是Ctrl键和“G”配合用时的特有功能。Ctrl键还有其它不同的组合，可得到各种不同的功能。这里暂不详述，以后会逐步提到。

2.3.5 Return键

这个键称为“回车”键，它是一条命令的执行键。从键盘上打入一串字符后，按下Return键，表示输入的内容已敲完。机器收到“回车”键后知道一条语句或命令已输入完，于是对输入的语句或命令进行分析和执行。在按回车键之前，若发现前面输入的字符有错，可以用退格键“←”和进格键“→”进行修改，修改好后再按Return键。在按过回车键之后再想修改可以使用屏幕编辑功能键。

2.3.6 进入选择方式（F6）

按下F6键后屏幕显示出系统固化的所有程序。用户敲入选择的序号，系统便进入该程序执行。通过F6键，用户可以较方便地切换屏幕显示方式，即可以在40列、80列及汉字方

式之间转换。通常情况下，0是进入40 列BASIC 程 序，1是进入汉字方式下的BASIC程 序，2是 进 入 80 列 方 式 下 的BASIC程序。3 是进入监控程 序。

2.3.7 自检（Ctrl-Reset-Test）

同时按下Ctrl键、Reset键和Test键，释放后（Test键最后抬起）系统进入自检操作。

在检测内存时，屏幕上显示被检测到的内存单元地址。当某一个内存区域检测结束时，显示出检测结果。当测试正确时显示“OK”。如果测试出不为所知的内容时显示“UNKNOWN”。若存储器不存在，则在相应位置显“NULL”。

当存储器测试完后，将进行彩色显示测试，屏幕显示出16种不同颜色的彩条。

自检程序可以连续循环地测试，并在TIMES一栏显示示测试的次数。如需调试彩色可在检测存储器时按任一键进入彩色显示测试，然后再按任一键使屏幕一直显示彩条，直到你调试完彩色电视机，再按任一键又进入自检程序循环测试。若正在检测彩色显示，则只要按任一键便可停在彩条显示状态，再按任一键进入自检程序循环测试。

如果要中断自检测试，可同时按下Ctrl-Reset键，机器便进入BASIC状态。

2.3.8 系统复位（Ctrl-Reset）

Reset键称为复位键。当你的程序在执行过程中机器失去控制，或者程序在执行了一半不希望它继续执行下去了。

此时，一种办法是关掉机器电源，不过这样就会失去内存中的程序和数据。另一种办法就是利用Reset键进行系统复位，这样可以不破坏内存中的程序。系统复位的具体做法是按下Ctrl键的同时再按下Reset键，按键释放以后就会中断机器的任何工作，系统重新进入BASIC，并把控制交给键盘。

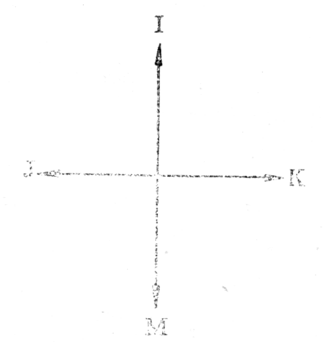
2.3.9 退格键和进格键（←，→）

“←”是退格键，或叫左箭号键。“→”是进格键，或叫右箭号键。利用这两个键可以修改输入过程中的各种错误。“←”键象打字机上的空格退位键一样，每按一次，光标退回一格，退回光标下的字符就从内存中抹去。例如我们先在键盘上打入一串字符，然后连续敲几下退格键，在屏幕上就会看到光标沿着刚打入的字符退位返回，但原来字符在屏幕上并不消失。然而，这些字符实际上已从计算机的存储器中抹去了。当光标退位退到左端时，若再按“←”键，则光标就跳到下一行，并产生一个新的提示符。

“→”键是使光标沿显示行向右移动（前进方向）。当按下此键时，光标沿着显示行向前移动，它所经过的每一个字符又重新复制到存储器里，就象重新按下该字符键输入一个新的字符一样。为了更清楚地了解“→”键的作用，可以在键盘上试一下。例如连续地打入一串字符，并用“←”键将光标返回几格，光标经过的字符便从存储器中抹去，然后再按几次“→”键，光标经过的字符又准确无误地送入了存储器。

2.3.10 编辑功能键（Esc）

Esc是英文ESCAPE的缩写。在通常情况下指“逃跑”“脱离”等意思，在计算机中指“换码”。它是一个控制字符，它表示在它后面的一个字符或一组字符应根据不同的代码或不同的编码字符集来解释。在中华学习机的BASIC或监控状态下，这个字符表示进入屏幕编辑状态。例如，先按一下Esc键，然后按几次“K”键，这时屏幕上并不会显示“K”字符，而是使光标向右移动几次，然后再按几次“M”键，这时光标向下移动，再按“I”键，则光标向上移动，最后再按“J”键，会看到光标向左移动。按了这4个键以后，会发现这4个键“K、M、I、J”在键盘上正好形成十字，分别代表4个方向，即：



只有在进入屏幕编辑状态时，即按过Esc键时，这4个键可以反复地用来改变屏幕上光标的位置。若要从编辑状态回到原来状态，只要按一下空格键或其它任何一个键即可。这就是Esc键的功能。它可以任意地将光标移到屏幕上的各种位置上。

Esc键提供的功能和进格键、退格键配合使用在调试程序时非常有用。例如，在调试程序时用LIST命令列出程序，当发现前面程序有错而光标此时已在下面时，只要按一下Esc键，然后利用I、J、K、M键把光标移到需要修改的语句行的第一个字符上。此时先按一下空格键或任何一个

键，退出编辑状态，接着，按进格键“→”，把光标移到需要修改的字符上即可，键入需要的字符之后，再用进格键使光标移到这条语句末尾的后一个位置，然后按回车键即可。

2.3.11 其它功能键

上箭号键（↑）：按一下光标上移一行。

下箭号键（↓）：按一下光标下移一行。

制表键（Tab）：如同打字机上的定位键。比如说屏幕定了1，11，21，31四个位置，光标在13的位置上，按Tab键就会移到下一个定位21上。

删除键（Delete）：在监控、BASIC状态下同退格键“←”一样。

冷启动键(Boot)：Control-Reset-Boot进入冷启动。

F1键：汉字系统字符输入方式。

F2键：汉字系统汉字输入方式。

F3键：汉字系统下造汉字。

F4键：汉字系统下造字符。

F5键：汉字系统下改变字符组。

Enter键：作用同Return键。

Pause键：暂停键。暂停显示输出，按任一键继续。

Break键：中止键。中断并退出程序的执行。

第三章　汉字系统的操作使用

3.1　进入和退出汉字系统

进入和退出汉字系统有两种方式。即键盘方式和程序方式。

键盘方式进入汉字系统的操作很简单。先敲F6键，使系统显示选择单，再敲入“1”选择汉字BASIC方式，于是进入汉字系统。

退出汉字系统也很方便。先敲F6键，使系统显示选择单。再敲入“0”或“2”，可进入40列BASIC或80列BASIC。

程序方式进入汉字系统可执行如下程序：

```
10  POKE  766, 1
20  PR#3
```

退出汉字系统可执行

```
100  PRINT CHR$ (17)
```

如果系统装入了DOS操作系统，则进入汉字方式可执行如下程序：

```
10  POKE 766, 1
20  PRINT CHR$ (4) ; "PR#3"
```

退出操作和无DOS时一样。

进入汉字系统后，屏幕显示为高分辨图形的第二页，内存的程序不受影响。

3.2 输入汉字的操作

进入汉字系统后，屏幕最底行为状态行。状态行左端显示出“字符：”，这表示当前的输入方式为字符输入方式，即敲入的键码按ASCII字符集解释。当需要输入汉字时，先敲入F2（或Ctrl-T），状态行显示出当前系统挂接的所有输入方法，由用户选择。

用户通过敲入输入法序号进入相应的输入方式。

3.2.1 拼音方式

当敲入“1”选择拼音方式后，状态行左端显示“拼音：”，这表示进入拼音输入方式。CEC-ⅠA中华学习机的拼音输入为全拼音输入方式，即输入汉字时将该汉字的音节的声母和韵母全部输入。计算机根据输入的音节在状态行提示出该音节的所有字，其中包括了该音节所有不同声调的字。操作者可以根据状态行上提示的汉字序号用数字键选择所需要的汉字。因此，只要学过汉语拼音的人就很容易学会全拼音输入方法。下面是全拼音输入汉字的方法。

① 第一键提示常用字：当你第一键敲入了汉字的声母以后，屏幕状态行上将显示出该声母相关音节中最常用的6个汉字。此时若有需要的汉字，只要敲入相应汉字的序号即可；若没有，则要继续敲入该汉字的韵母。

② 向后或向前寻找同音节字：当敲完汉字的韵母后，状态行提示出6个同音节字。若出现所需要的汉字，即可敲入相应的汉字序号；若这6个同音节字中没有要输入的汉字，则可按“>”键继续向后寻找。每按一次“>”键，屏幕状

态行就显示出下一幕6个同音节字，直到机器发出“嘟”的一声，表示该音节的汉字已全部显示完。当要寻找前一幕汉字时，可按下“<”键，此时，前一幕的6个汉字便显示在状态行上。

③ 敲入所选汉字的序号：当屏幕状态行出现要输入的汉字时，只要敲入该汉字前面的序号，该汉字就会显示在屏幕当前光标的位置上，且该汉字的内码被输入到键盘输入缓冲区。连续敲入序号，可连续输入汉字。

④ 删除拼音提示：若输入汉字时，有拼音敲错，可直接按“Delete”键或者“←”键，使状态行上最后一个拼音字母被删除，或者按除号键“／”，使所有提示字符都删除。当状态行上的拼音字母都删掉以后，再按“Delete”键或“←”键则删除光标上的信息。

3.2.2 区位方式

当敲入“2”选择区位码输入方式后，状态行左端显示“区位：”，表示进入区位码输入方式。

进入区位输入方式以后，将使用国家标准GB2312规定的区位方法输入汉字。在输入汉字时，仅使用10个数字键即0～9，每4位数字对应一个汉字。例如国标中规定16区01位置的汉字是“啊”，键入1601后，“啊”字就显示在当前光标的位置上。其中前两位数字表示区号，后两位数字表示位号。国标一级汉字在16～55区，二级汉字在56～87区。一级字是按汉语拼音排序，二级字是按偏旁部首排序。因此我们可以通过查阅区位码表寻找出汉字的区位码来输入汉字。

当敲错区位码时，可按“Delete”键或“←”键，使区位码提示的最后一个数字被删除，当区位码都删掉后，再按

"Delete"键或"←"键，则删除光标位置上的信息。

3.2.3 特殊符号的输入

在拼音方式下，键入－（减号）、＝（等号）和＼（斜线）可以选择输入标点符号、算术运算符号和制表线等。这些符号的内容说明如下：

按"－"（减号）可输入的符号如图3-1。



图 3-1

按"＝"（等号）可输入的符号如图3-2。

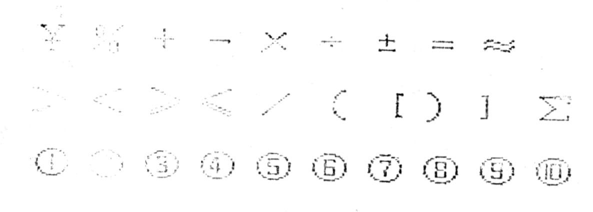


图 3-2

按"＼"（斜线）可以选择输入的如图3-3。

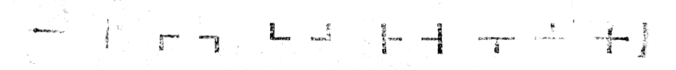


图 3-3

3.2.4 五笔字型输入法

五笔字型输入法是中华学习机的基本形码输入法之一。这一技术根据字根的组字频度、实用频度，将优选出来的一百多个字根，依其中的规律性、相容性、谐调性，分为5个区，每区5个位，共计25组字根，对应于英文键盘的25个字母键，形成了一个完全根据汉字的字形便可向电脑输入汉字和词语的高效率汉字输入法。初等文化水平的人，1至2天即可学会全部汉字和词语的编码，经指法训练，每分钟可输入120～180个汉字。

一、拼形组字法

人们常说

“木、子”——李

“日、月”——明

“文、刂”——刘

以及

“双　木”——林

“三　人”——众

“三　石”——磊

可见，汉字可用几个基本的部分拼合而成。这些象积木块一样，用来拼字的基本部分，叫做“字根”，能组成尽可能多的常用字的字根，叫做“基本字根”。把字根安排在键盘上，就形成了“字根键盘”，通过按键，就可以拼合出汉字。

例 1　S键上有“木”，B键上有“子”，按SB（再加按一下空格键）就组出了“李”。

例 2　K键上有“口”，按KKK（再按一下空格键）

就组出了“品”。

例 3 J键上有“日”，V键上有“刀”，K键上有“口”，O键上有“灬”，按JVKO就组出了“照”。

例 4 G键上有“王”，F键上有“土”，W键上有“亻”，R键上有“扌”，按GFWR就组出了“环境保护”（即打每一字的第一部分）。

这个过程叫“拼形组字（词）”。据此，五笔字型可把成千上万的汉字和词语输入计算机，叫做“拼形输入”。

二、汉字的五种笔画

书写汉字时，一次不间断地连续写成的一个线段，叫做

表 3-1

代号	笔划名称	笔划走向	笔划及其变形
1	横	左→右	一 ㇀
2	竖	上→下	丨 亅
3	撇	右上→左下	丿
4	捺	左上→右下	㇏ 丶
5	折	带转折	乙 ㇇ ㇕ ㇆ ㇌ ㇄ ㇙ ㇗ ㄣ

说明：

1. 由“现”是“王”字旁可知，提笔应属于横。
2. 由“村”是“木”字旁可知，点笔应属于捺。
3. 由旧体的“木”竖笔带钩可知，竖左钩应属于竖。
4. 一切带拐弯的笔划，都归为折类。

汉字的笔画。

汉字的笔画，不能切断，比如不能把“口”拆成“丨一丨一”。

汉字的笔画，按书写走向，可分为下表所示的五种，这里分别依其使用频度，命以代号1、2、3、4、5（见表3-1）。

三、字根键盘

汉字由字根组成，字根由笔画构成。笔画、字根、整字

五笔字型编码方案字根区位表

区号＼位号	1	2	3	4	5
横 1	王 一	土 二	大 三	木	工
竖 2	目 丨	日 ‖	口 川	田	山
撇 3	禾 丿	白 〃	月 彡	人	金
捺 4	言 丶	立 冫	水 氵	火 灬	之
折 5	已 乙	子 《	女 巛	又	纟

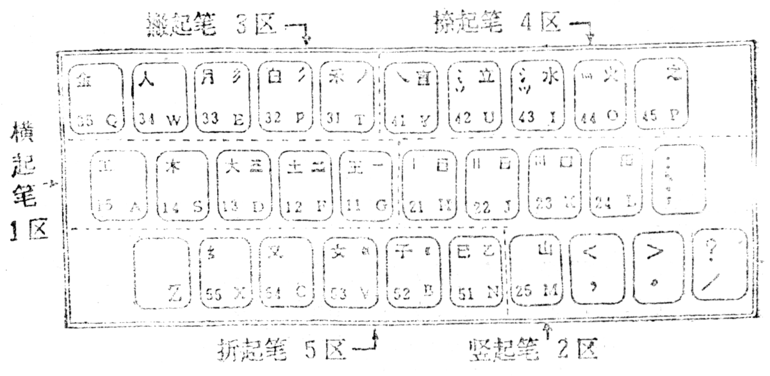


图 3-4 键盘分区及键位安排图

是汉字结构的三个层次。将一百多种基本字根按首笔笔画分作五类，各对应英文键盘上的一个区，每区又分作五个位，位号从键盘中部向两端放射排列。共5×5＝25个键位。各键位的代码，既可以用区位号（11—55）表示，也可用对应的英文字母表示。键盘分区及键位安排情况如图3-4。

四、字根总表

把全部字根都标记在键盘上，就成了5区25位的字根总表（见图3-5）。同一键位上字根，都使用同一个代号，可分为四种类型。

1．首笔与区号一致，次笔与位号一致，如

王：在1区1位；　　白：在3区2位；
石：在1区3位；　　文：在4区1位；
之：在4区5位；　　纟：在5区5位。

2．首笔符合区号，且笔画数目及其外形与位号相符。如：

三：在1区3位；　　水：在4区3位；
女：在5区3位；　　日：在2区2位。

3．与主要字根形态相近或渊源一致。如

“⺌”、“氺”、“⺗”属于“水”
“禾”在“禾”键上
“耳”在“卩”键上
“扌”在“手”键上

4．个别例外：笔划特征与所在区、位号不相符合，同时与其它字根之间又缺乏联想性的字根计有：

“车”、“力”在“24．L”键上(繁体“车”与“甲”相似，“力”的声母为“L”)

五笔字型键盘字根总图

35 Q	34 W	33 E	32 R	31 T	41 Y	42 U	43 I	44 O	45 P
15 A	14 S	13 D	12 F	11 G	21 H	22 J	23 K	24 L	： ；
Z	55 X	54 C	53 V	52 B	51 N	25 M	< ，	> 。	？ /

五笔字型字根助记词

11 王旁青头戋（兼）五一，
12 土士二干十寸雨。
13 大犬三𦍌(羊)古石厂，
14 木丁西，
15 工戈草头右框七。

21 目具上止卜虎皮，
22 日早两竖与虫依。
23 口与川，字根稀，
24 田甲方框四车力。
25 山由贝，下框几。

31 禾竹一撇双人立，反文条头共三一。
32 白手看头三二斤，
33 月彡（衫）乃用家衣底。
34 人和八，三四里，
35 金勺缺点无尾鱼，犬旁留乂儿一点夕，氏无七（妻）。

41 言文方广在四一，高头一捺谁人去。
42 立辛两点六门疒，
43 水旁兴头小倒立。
44 火业头，四点米，
45 之宝盖，摘礻（示）衤（衣）。

51 已半巳满不出己，左框折尸心和羽。
52 子耳了也框向上。
53 女刀九臼山朝西。
54 又巴马，丢矢矣，
55 慈母无心弓和匕，幼无力。

图 3-5

“心”在“51. N”键上（“心”字最长的笔划为折笔）。

五、怎样找到字根

1. 王：首笔横，故在1区，次笔横，故在1位(1 1、G)

2. 又：首笔折，故在5区，次笔捺，故在4位(5 4、C)

3. 雨：首笔横，故在1区，次笔竖，故在2位(1 2、F)

4. 宀：首笔点，故在4区，次笔折，故在5位(4 5、P)

5. 三：首笔横，故在1区，三个横，故在3位(1 3、D)

6. 灬：首笔点，故在4区，四个点，故在4位(4 4、O)

7. 口：首笔竖，故在2区，可在2区的五个键中找到

8. 耳：从“阝、卩”，首笔折，次笔竖，故在（52、

9. 力：属例外，“力”声母为“L”，即“24”键。

10. 立六辛疒冫丬丷亠等字根中均有两点，都在4区2位。

11. 田口四皿等“四方”形字根，首笔竖，“四方”，在“24”

六、五笔字型编码流程图

五笔字型技术，将全部汉字分为两大类：

① 键面字：指键面上的全部字根；

② 键外字：指一切键面字根以外的字。

对汉字依形编码的总方针是：键外字拆成字根，字根拆成单笔画。现分别举例如图3-6。

七、键面字的编码输入

1. 键名汉字输入

各键位左上角的黑体字根，叫“键名”。如王、日、口等，共计25个，它们的输入方法是把所在键连击四下。

王：11 11 11 11 (GGGG)

五笔字型汉字编码流程图

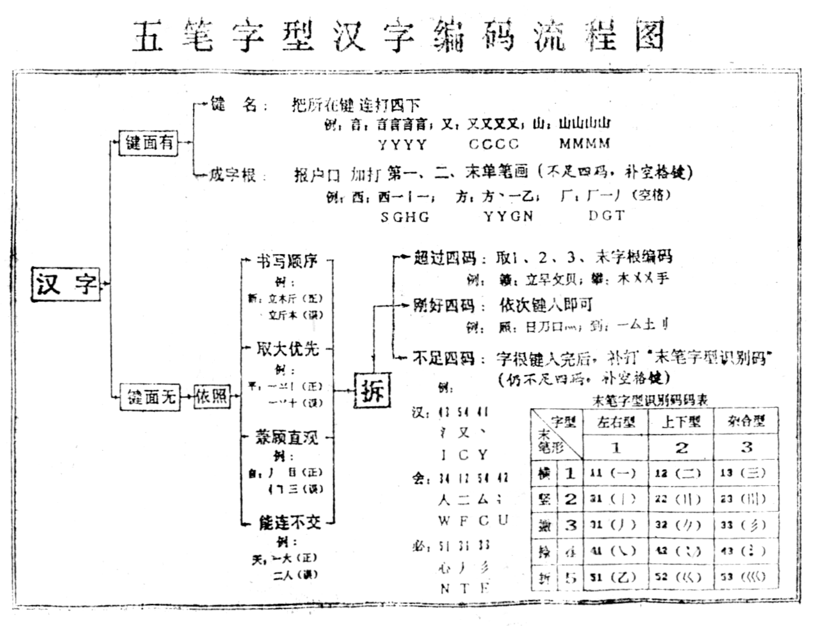


图 3-6

大：1 3　1 3　1 3　1 3（DDDD）

之：4 5　4 5　4 5　4 5（PPPP）

言：4 1　4 1　4 1　4 1（YYYY）

纟：5 5　5 5　5 5　5 5（XXXX）

2. 成字字根输入

字根总表中，键名以外、本身即是汉字（包括“亻”、“氵”等有国标码的部首在内）的字根，叫“成字字根”。其输入方法为：

先打该字根所在的键一下（报户口），再打该字根的第一、第二及最末一个单笔画（“报户口”后一定是单笔画），如：

方：方（报户口）　丶（首笔）　一（次笔）　乙（末笔）

　4 1. Y　4 1. Y　1 1. G　5 1. N

石：石（报户口）　一（首笔）　丿（次笔）　一（末笔）

　1 3. D　1 1. G　3 1. T　1 1. G

用：用（报户口）　丿（首笔）　乙（次笔）　丨（末笔）

　3 3. E　3 1. T　5 1. N　2 1. H

力：力（报户口）　丿（首笔）　乙（次笔）

　2 4. L　3 1. T　5 1. N　（空格）

干：干一一丨（1 2　1 1　1 1　2 1　FGGH）

厂：厂一丿　（1 3　1 1　3 1　DGT　）

十：十一丨　（1 2　1 1　2 1　FGH　）

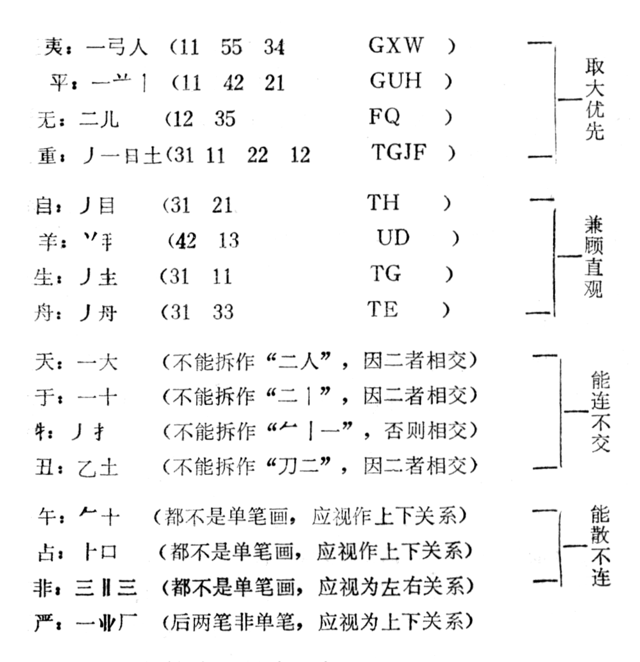
氵：氵丶丶一（4 3　4 1　4 1　1 1　IYYG　）

八、键外字的编码输入

字根总表中没有的笔画结构，均应按书写顺序，依次拆成为总表中已有的最大字根，以增加一笔不能形成已有的最大字根，来决定笔画分组，直到把整个汉字拆分完毕。

拆分原则可归纳为以下四个要点：

取大优先，兼顾直观，能连不交，能散不连。



九、汉字的字型和识别码

鉴于“叭”、“只”、“旮”、“旭”这样的字，单凭字根不能区分，有必要采用字型信息（“五笔字型”的“型”源于此）。

鉴于“沐”、“汀”、“洒”三字中，“木”、“丁”、“西”在同一个键上，引起了重码。但这三个字的末笔画不同，故可用最后一个笔画加以区分。

将以上两者结合起来，就是汉字的“末笔字型交叉识别”码。

1．汉字的三种型

根据构成汉字的各字根之间的位置关系，可以把成千上

万的方块汉字分为三种类型：1. 左右型，2. 上下型，3.杂合型，如表3-2所示。

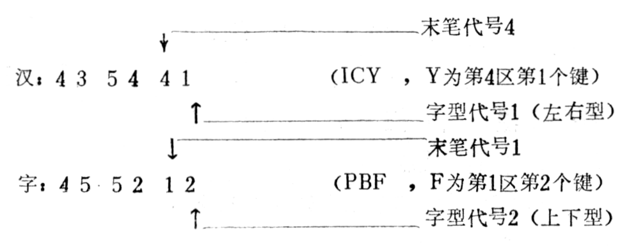
表3-2

字型代号	字型	图示	字例
1	左右		汉 湘 结 封
2	上下		字 莫 花 华
3	杂合		困 凶 这 司 乘 本 重 天 且

2. 末笔字型交叉识别

“末笔字型交叉识别”只适用于不足四个字根组成的字。

对于拆不够四个字根的汉字，为了避免使用上述“Z”键及在提示行中挑选，有必要在字根打完后，加上一个末笔字型交叉识别码，识别码由末笔画代号与字型代号组合而成，如：



↓———————— 末笔代号2
华：34 55 12 22 （WXFJ，J为第2区第2个键）
↑———————— 字型代号2（上下型）

↓———————— 末笔代号1
同：25 11 23 13 （MGKD，D为第1区第3个键）
↑———————— 字型代号3（杂合型）

↓———————— 末笔代号1
本：14 11 13 （SGD ，D为第1区第3个键）
↑———————— 字型代号3（杂合型）

↓———————— 末笔代号4
东：15 43 43 （AII ，I为第4区第3个键）
↑———————— 字型代号3（杂合型）

注意事项：

①“键名”及一切成字字根都不再用识别码，如：

厂：厂一丿（13 11 31 DGT），虽不足四码也不用识别。

② 如果一个字加了识别码后仍不足四码，则必须打空格键。

③ 为了有足够的区分能力，对于“进”、“连”这样带“走之”的字，它的“末笔”规定为被包围部分的末笔：

进：二川辶（12 22 45 23 FJPK）

④ 对于习惯笔顺不一致的“刀”、“力”、“九”、“匕”四个字根，当它们参加“识别”时，一律规定用“折笔”作末笔。如：

花：艹亻匕（15 34 55 52 AWXB）

十、万能学习键“Z”

"Z"键为万能学习键，它不但可以代替"识别码"，帮你把字找出来，告诉你"识别码"，而且，还可以代替你一时记不清或分解不准的任何字根，并通过提示行，使你知道"Z"键对应的键位或字根。

例如，你要敲入"越"字，已知左边是"土"和"止"，因而敲入"F"和"H"，但"戉"不知怎么敲，这时可以打入"Z""Z"，就像代数一样，先用两个"Z"代替那两个不知道的键。这时状态行显示出"fhAG 1墟"，如果按一下"]"键，状态行显示出：fhAT 1越"于是，你知道了，后两键是"AT"。如果再多次按下"]"键，可以从状态行看到所有前两键是"fh"的字，且后两键也以大写的形式出现在"fh"后。

当打四个"Z"时，即可查阅全部汉字字库中的字及外码。

十一、简码输入

常用汉字中，多数可只取其前边的一至三个字根，再加空格输入。即只取该一汉字全码的最前边一个、二个或三个字根（码）输入，形成所谓一、二、三级简码。

一级简码（即高频字码）：

一：	11（G）	要：	14（S）
的，	32（R）	和：	31（T）

二级简码：

化，	亻匕（WX）	信：	亻言（WY）
李：	木子（SB）	张：	弓丿（XT）

三级简码：

华：	亻匕十（WXF）	规：	二人贝（FWM）
陈：	阝七小（BAI）	得：	彳日一（TJG）

有时，同一个汉字可有几种简码。例如"经"，就有

一、二、三级简码及全码等四个输入码。

经：55（X）　　　　　　经：55 54 15（XCA）

经：55 54（XC）　　　　经：55 54 15 11（XCAG）

十二、词语输入

① 两字词：每字取其全码的前两码组成，共四码。

经济：纟又氵文（55 54 43 41 XCIY）

操作：扌口亻𠂉（32 23 34 31 RKWT）

② 三字词：前两字各取一码，最后一字取两码，共四码。

现代化：王亻亻七（11 34 34 55 GWWX）

操作员：扌亻口贝（32 34 23 25 RWKM）

③ 四字词：每字各取全码的第一码

中华民族：口亻尸方（23 34 51 41 KWNY）

聪明才智：耳日十丿（52 22 12 31 BJFT）

十三、重码及容错码

有相同编码的字叫"重码字"。如键入DYI，即显示：

1太2丈

如需要"太"，就不必挑选，敲入空格键后只管输入下文，"太"就会自动显示到现行光标位置上来，如要的是"丈"字，可打上排数字键"2"。

由于CEC-IA型中华学习机内存的限制，输入程序没有设计容错码和字根区位输入。

十四、键盘操作

① 大写的26个字母为英文字母直接输入系统。

② 小写的26个字母作为五笔形输入码。其中Z键为学习键。

③ 用“[”和“]”键使状态行翻页。

④ 用数字键“1”～“6”选择输入提示字。

⑤ 用数字键“0”清除全部输入键。

⑥ 用“←”或“Delete”键清除刚敲入的一个输入码。若提示行已删完则删文本区字符。

⑦响铃表示有重码或按键无效或操作失败。

⑧用数字键“7”～“9”输入汉字库中前9区符号。“7”、“8”、“9”分别表示前三区（1、2、3区）、中三区（4、5、6区）和后三区（7、8、9区）。用“A～Y及a～u”对每三区分段，与翻页键和数字选择键配合使用可输入所需的字符。

⑨ 空格键的使用：

〈1〉当外码长度为1时，按空格键可输入高频字25个。

〈2〉当外码长度为2时，按空格键可输入二级字625个。

〈3〉当外码长度为3时，按空格键可输入提示行中第1个提示字。

3.2.5 笔形码输入法

笔形码输入法是中华学习机的基本形码输入法之一。下面是它的基本输入方法介绍。

一、笔形编码

1. 笔形可分为八种，用八个号码来代表（表3-3）。

表 3-3

代号	1	2	3	4	5	6	7	8
名称	横	竖	撇	点	折	弯	叉	方
笔形	一 ㇀	丨 亅	㇒ ㇒	丶 ㇏	㇕ 亅	㇄ ㇂	十 ㄨ	口 口

2. 按笔画的位置次序编码（不管笔顺），先上后下，先左后右。

小	早	禾	半	右
534	817	3734	7431	318

3. 把合体字按上下、左右、外内分成两部分，每部分各取三码，全字最多取六码。

盘	他	壮	装	病
335-252	32-265	241-71	241-413	413-132

（本编码法备有按音序检索的笔形码本，遇有疑义字，可通过字音查到该字准确的笔形编码。）

二、屏幕引导

为向操作人员指示文字编码，在显示屏上专门开设了一个编码引导区，电脑进入笔形码输入方式时，尚未击键，显示屏便出现若干字，其上各标号码。例如：

1	2	3	4	5	6	7	8
一	上	的	这	了	乙	十	是

击其上的号码键，码下的字即进入输入状态（这八个字的编码即其最高笔画的代码），例如要输入“的”字，可按3键，这时引导区画面变成：

	1	2	3	4	5	6	7	8
的	在	他	行	人	力	九	千	自

如不输入“的”字，而要输入“他”字，继续按2键，这时画面变成：

1	2	3	4	5	6	7	8
他	便	川	作	们	传	亿	什	但

如不输入“他”字，而要输入“便”字，再按1键……如此类推，每输入一个字以后，引导区画面立即恢复到刚进入笔形码输入方式的初始状态。

因此：“的”字输入一码3即可；

“他”字输入两码32即可；

“便”字输入三码321即可。

三、操作键盘

在CEC-ⅠA机上笔形码的操作键盘规定如下(使用键盘右边的小键盘区)：

“1～8”键为笔形号码。

“0”键将左端提示字送到光标处。

“9”键将提示的双字词送到光标处，如不是所需要的词仍按0键则将单字送到光标处。

“·”键用于有重码时的翻页键。当提示行中出现反相显示的数字时，该数值即表示重码数。翻页后可显示全部重码字（最多九个字），按数字键，可将所需重码字选中。

“+”键可输入如下所示的9个符号（等效于敲入“1”“4”两键）：

1	2	3	4	5	6	7	8
，	、	：	？	！	；	…	—	·

“-”键可输入如下所示的9个符号（等效于敲入“2”“7”两键）：

	1	2	3	4	5	6	7	8
·	“	”	‘	’	(	)	《	》

“Delete”键和“←”键都可用来删除已输入的笔形编码，若状态行笔形编码提示删除完继续按“Delete”键或“←”键，则删除文本区光标处的信息。

敲入“4”“8”两键后再按正常笔形编码顺序输入编码，可得到常用的偏旁部首。如：

亅　忄　辶

483　48244　4854

3.3 多字符组的输入

CEC-ⅠA中华学习机具有使用多种字符组的功能。它允许用户在屏幕上同时显示10种不同字符集的字符，为应用软件的编写提供了极大的方便。我们把ASCII字符集作为基本字符组，把IBM-PC机上扩充的图形符号作为扩充字符组。为了适应教学软件的需要，系统增加了角标字符组和造字字符组。

字符组的选择是通过敲入F5键后敲入一个数字键实现的。这个数字键就是所选字符组的组号。

字符组的分配情况如下：

0 组：基本字符组

1 组：扩充字符组

2 组：上角标字符组

3 组：下角标字符组

4 组：左上角标字符组

5 组：左下角标字符组

6 组～9组：造字字符组

下面详细说明各组的使用方法。

3·3·1　基本字符组

进入汉字系统时，系统使用基本字符组。基本字符组共128个。0～31为控制字符，32～126为可显示字符，127为汉字标识符。用下述程序可将基本字符组显示并打印出来。

```
10  POKE  963, 3
20  POKE  2043, 50
30  FOR I=32 TO 126
40  PRINT  CHR$ (I) ; " ";
50  NEXT
60  POKE  963, 3: PRINT
70  POKE  963, 1
```

其中10、20、60和70语句是用来控制打印的，若是仅显示基本字符组，可以只运行30、40和50语句。图3-7是运行的结果。

```
  ! " # $ % & ' ( ) * + , - . / 0 1
 2 3 4 5 6 7 8 9 : ; < = > ? @ A B
C D E F G H I J K L M N O P Q R S T
 U V W X Y Z [ \ ] ^ _ ` a b c d e
f g h i j k l m n o p q r s t u v w
 x y z { | } ~
```

图 3-7

对于基本字符组的前32个控制字符，当设置了不处理控制字符标置后，可以显示和打印出它们对应的图形符号。下

面的程序将显示并打印出包括前32个图形符号的基本字符组。

```
10  POKE  963, 67
20  POKE  2043, 50
30  FOR I=0 TO 126
40  PRINT  CHR$ (I) ; " ",
50  NEXT
60  POKE  963, 3: PRINT
70  POKE  963, 1
```

运行结果如图3-8。

```
# $ % & ' ( ) * + , - . / 0 1 2 3 4
5 6 7 8 9 : ; < = > ? @ A B C D E
F G H I J K L M N O P Q R S T U V W
X Y Z [ \ ] ^ _ ` a b c d e f g h
i j k l m n o p q r s t u v w x y z
{ | } ~
```

图 3-8

程序中使用了963（$3C3）单元。此单元的第6位为1时，系统不处理控制字符，并把控制字符作为可显示字符一样处理。963单元的详细介绍可参阅第3.5节显示控制操作。

3.3.2 扩充字符组

输入扩充字符组的方法是，敲入F5键，再敲数字键1。扩充字符组有127个图形字符，对应的内码是0～126。127是汉字标识符。用下面的程序可以显示并打印出扩充字符组的全部字符。

```
10  POKE  963, 67
15  POKE  966, 1
20  POKE  2043, 50
30  FOR I=0 TO 126
40  PRINT  CHR$ (I);
45  POKE  966, 0: PRINT"  "; : POKE 966, 1
50NEXT
55  POKE  966, 0
60  POKE  963, 3: PRINT
70  POKE  963, 1
```

运行结果如图3-9。

图 3-9

程序中使用了966单元。这个单元是存放当前字符组号的。语句15就是把字符组设置为1号字符组。通常，在使用过非零号字符组后应回到零号字符组。因此程序中55语句又把字符组恢复为0号字符组。

3.3.3 角标字符组

角标字符组共有四个。它们的区别仅是显示的位置不同，而图形符号是一样的。上角标字符位于基本字符组字符的上半个字符位置。例如A^X中的X就是上角标字符。下角标字符位于基本字符的下半个字符位置。例如A_Y中的Y就是下角标

字符。由于每输入一个字符，光标就向右移动一个字符位置，因此要输入 A_{Y}^{X}这样同时有上下角标的字符串就要使用左上角标和左下角标。例如，当输入完A^{X}时光标已走到X字符的右边即A^{X}_，这时换用左下角标字符组，即输入字符位于当前光标的左边，再输入Y就得到 A_{Y}^{X}_。注意，这时要立刻把字符组换到基本字符组，即0组。输入A_{Y}^{X}的过程是敲入下面一串字符：

$$AF_5 2XF_5 5YF_5 0$$

下面的程序显示并打印出角标字符组的所有字符，不包括127，因为它是汉字标识符。

```
10  POKE 963, 67
15  POKE 966, 2
20  POKE 2043, 50
30  FOR I=0 TO 126
40  PRINT CHR$ (I) ; " ";
50  NEXT
55  POKE 966, 0
60  POKE 963, 3: PRINT
70  POKE 963, 1
```

运行结果如图3-10。

```
# $ % & ' ( ) * + , - . / 0 1 2 3 4
5 6 7 8 9 : ; < = > ? @ A B C D E
F G H I J K L M N O P Q R S T U V W
X Y Z [ \ ] ^ _ ` a b c d e f g h
i j k l m n o p q r s t u v w x y z
{ | } ~
```

图 3-10

在使用了角标字符组后，可以输入和输出各种数学表达式，例如下面的程序，

```
10  PRINT"A²+B²=C²"
15  PRINT
20  PRINT "log₂(x^Y)=? "
25  PRINT
30  PRINT "A₁^X * A₁^Y= A₁^(X+Y)"
```

运行结果如下：

$A^2+B^2=C^2$

$\log_2(X^Y)=?$

$A_1^X * A_1^Y=A_1^{X+Y}$

如果用户需要使用的字符在前面介绍的各种字符组中都没有，则可以使用造字字符组。

3.3.4 造字字符组

系统规定第6、7、8、9四个字符组为造字字符组。造字字符组的字符完全由用户自己确定。用户可以通过系统提供的造字软件造出16×8点阵的字符，并通过选择相应的字符组号显示和打印出所造的字符。例如，图3-11中的手写体HELLO就是利用造字字符组输入和显示出来的。

```
10  PRINT "Hello"

]RUN
Hello
```

图 3-11

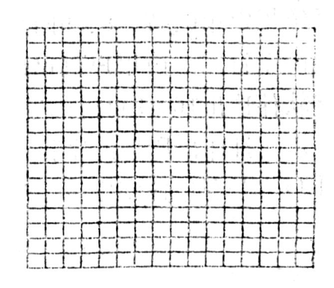
3.4 造字操作

CEC-ⅠA中华学习机在系统程序中提供了造字功能。有了造字功能，用户可以根据自己的需要造出汉字字库中没有的汉字，可以造出任意16×16点阵的图形和16×8点阵的图形。

3.4.1 造中文字

在进入汉字系统后，敲入F3键，系统进入造字程序。屏幕显示出16×16的方格图形。如图3-12所示。方格左面是一些说明，在后面详细说明，先看方格下面的状态行。状态行此时提示用户输入所造字的区位码。从提示中可看到，区号的选择应在10和15之间。即所造字的区号在国标区位码的10

上:↑ 下:↓
左:← 右:→
存字:ctrl-s
清字:ctrl-c
退出:ctrl-q
画点/去点:空格



造字: 输入区号(10~15)位号(01~94)?

图 3-12

区和15区之间；位号的选择应在1～94之间。这时用户应连续敲入4个数字，前两个是区号，后两个是位号。例如选10区1位，输入1001，这时屏幕状态行提示你输入的区位号，并提问是否正确。如图3-13所示。此时若用户因疏忽敲错了区位号可以敲入“N”，并重新输入区位号。若刚才的输入正确，敲入“Y”，于是屏幕显示出新的状态提示，询问用户是否要输入参考字区位号。如图3-14。如果用户希望在一个已有汉字的基础上经过修改得到一个新的汉字，就可以在此时输入这个已有字的区位号。例如要利用已有字的偏旁部首或在已有字上添加笔画，都可以把这个已有字作为参考字。例如所造字需要有“鱼”字偏旁，可输入8713得到“鳝”字点

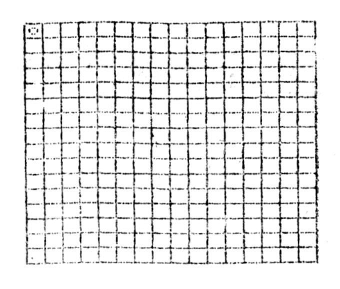


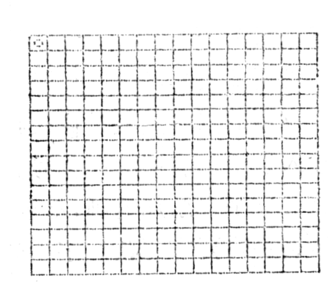
图 3-13

阵，如图3-15。将其中“善”字换成用户所希望的笔画就可以造出用户所需要的字，而“鱼”字偏旁就可以不用再画

了。

如果用户不需要参考字，则输入“N”，这时，方格中显示出所选区位号对应的内存单元的点阵。如果开机后第一

上:↑ 下:↓
左:← 右:→
存字:ctrl-s
清字:ctrl-c
退出:ctrl-q
画点/去点:空格



造字：输入参考字区位号?

图 3-14

上:↑ 下:↓
左:← 右:→
存字:ctrl-s
清字:ctrl-c
退出:ctrl-q
画点/去点:空格

造字：

图 3-15

次造这个区位号的字，内存中的点阵是随机状态的，如图3-16所示。此时，只要敲入Ctrl-C，便可将方格中的点都清除掉。若已对这个区位号造过字，则上一次造的点阵会直接显示到方格中。在图3-15中，方格下面有一个小的鳍字，这个字就是所造字在实际显示时的效果。

输入了所造字的区位号和参考字区位号后便进入方格中点阵的修改。这时可参考方格左面的说明。键盘右下方有四个箭号键。通过这四个键的使用可以移动方格中的光标。空

上：↑　下：↓
左：←　右：→
存字：ctrl-s
清字：ctrl-c
退出：ctrl-q
画点/去点：空格

造字：

图 3-16

格键使光标所在格中的内容变反。即原来有点则按空格键后消去点，原来没点按空格键后画上点。当所造的字完成之后，按Ctrl-S（同时按下Ctrl键和S键）即可将点阵存入内存。此时状态行又提示用户输入下一个所造字的区位号，回到图3-12所示的状态。若希望退出造字程序，按Ctrl-Q键即可（同时按下Ctrl键和Q键）。

3.4.2 造西文字

在汉字系统下敲入F4键，系统进入造西文字符的程序。首先屏幕显示出16×8的方格图形。状态行提示用户输入字符组号和字符。如图3-17。此时用户应输入字符组号和所造

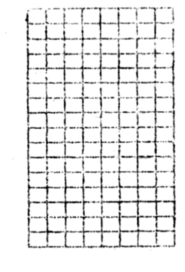


图　3-17

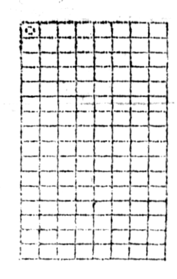
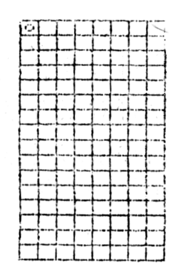


图　3-18

字符对应的键。字符组号应在6和9之间。例如选6字符组的K键，应输入“6K”，这时状态行提示输入正确否。如图3-18。若输入有错可在此时敲“N”然后重新输入。若输入正

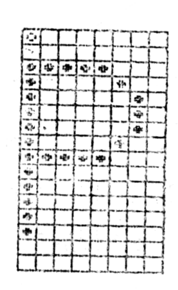
上:↑ 下:↓
左:← 右:→
存字:ctrl-s
清字:ctrl-c
退出:ctrl-q
画点/去点:空格



造字: 输入参考字组号和字符?

图 3-19

上:↑ 下:↓
左:← 右:→
存字:ctrl-s
清字:ctrl-c
退出:ctrl-q
画点/去点:空格



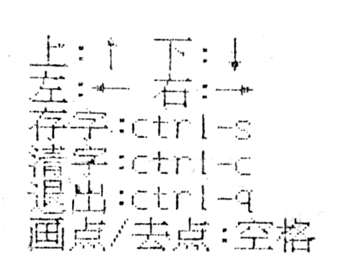
P

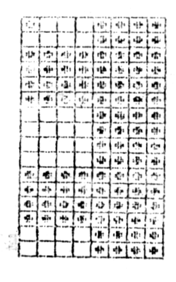
造字:

图 3-20

确可敲“Y”，于是状态行提示用户输入参考字组号和字符。如图3-19。这时用户可以输入一个参考字符，即用户希望在一个已有字符的基础上进行修改。例如，输入参考字符组号为0，字符为P，可敲入“0P”，此时屏幕显示如图3-20。

若用户不想使用参考字可敲入“N”，此时，方格中显示出所造字符组中字符对应内存单元的点阵。如果开机后第一次造这个字符，内存中的点阵是随机状态的，例如，图3-21。





造字：

图 3-21

此时只要敲入Ctrl-C，便可将方格中的点都清除掉。若已对这个字符组的该字符造过字，则上一次造的点阵会直接显示到方格中。

在输入了字符组号和字符以及参考字符组号和参考字符后便可以修改方格中的点阵了。这时可参考方格左面的说明。因前一节已作了说明，这里不再重复。与前一节不同的是按下Ctrl-S后，屏幕回到图3-17，提示用户输入下一个所

造字符的组号和字符。

3.5 汉字字串与ASCII字串的处理

进入汉字系统后，键入的ASCII字符占用一个字节。其代码为 $01～$7E。

键入的汉字均以等长的3字节码保存在主机的内 存 中。

表3-4 区位码、国标码、内码对照表

区位码	国标码	异形国标码	学习机内码		区位码	国标码	异形国标码	学习机内码	
10进制	16进制	16进制	16进制	10进制	10进制	16进制	16进制	16进制	10进制
1	21	A1	1D	29	20	34	B4	32	50
2	22	A2	1E	30	21	35	B5	33	51
3	23	A3	1F	31	22	36	B6	34	52
4	24	A4	20	32	23	37	B7	35	53
5	25	A5	21	33	24	38	B8	36	54
6	26	A6	23	35	25	39	B9	37	55
7	27	A7	24	36	26	3A	BA	38	56
8	28	A8	25	37	27	3B	BB	39	57
9	29	A9	26	38	28	3C	BC	3B	59
10	2A	AA	27	39	29	3D	BD	3C	60
11	2B	AB	28	40	30	3E	BE	3D	61
12	2C	AC	29	41	31	3F	BF	3E	62
13	2D	AD	2A	42	32	40	C0	3F	63
14	2E	AE	2B	43	33	41	C1	40	64
15	2F	AF	2D	45	34	42	C2	41	65
16	30	B0	2E	46	35	43	C3	42	66
17	31	B1	2F	47	36	44	C4	43	67
18	32	B2	30	48	37	45	C5	44	68
19	33	B3	31	49	38	46	C6	45	69

续表3-4

区位码	国标码	异形国标码	学习机内码		区位码	国标码	异形国标码	学习机内码	
10进制	16进制	16进制	16进制	10进制	10进制	16进制	16进制	16进制	10进制
39	47	C7	46	70	67	63	E3	62	98
40	48	C8	47	71	68	64	E4	63	99
41	49	C9	48	72	69	65	E5	64	100
42	4A	CA	49	73	70	66	E6	65	101
43	4B	CB	4A	74	71	67	E7	66	102
44	4C	CC	4B	75	72	68	E8	67	103
45	4D	CD	4C	76	73	69	E9	68	104
46	4E	CE	4D	77	74	6A	EA	69	105
47	4F	CF	4E	78	75	6B	EB	6A	106
48	50	D0	4F	79	76	6C	EC	6B	107
49	51	D1	50	80	77	6D	ED	6C	108
50	52	D2	51	81	78	6E	EE	6D	109
51	53	D3	52	82	79	6F	EF	6E	110
52	54	D4	53	83	80	70	F0	6F	111
53	55	D5	54	84	81	71	F1	70	112
54	56	D6	55	85	82	72	F2	71	113
55	57	D7	56	86	83	73	F3	72	114
56	58	D8	57	87	84	74	F4	73	115
57	59	D9	58	88	85	75	F5	74	116
58	5A	DA	59	89	86	76	F6	75	117
59	5B	DB	5A	90	87	77	F7	76	118
60	5C	DC	5B	91	88	78	F8	77	119
61	5D	DD	5C	92	89	79	F9	78	120
62	5E	DE	5D	93	90	7A	FA	79	121
63	5F	DF	5E	94	91	7B	FB	7A	122
64	60	E0	5F	95	92	7C	FC	7B	123
65	61	E1	60	96	93	7D	ED	7C	124
66	62	E2	61	97	94	7E	FE	7D	125

其格式为：

7F＋区码＋位码

由于在BASIC程序中，所有字符串在存贮时最高位被屏蔽，其代码值在 $00～ $7F之间，同时，在INPUT读入变量时，需要使用逗号（，）、冒号（：）和引号（″″）的代码来区分变量及判断字符串的结束。因此，这里的区位码不能完全和国标中的代码一样。这里的区位码是通过转换的，它与国标中规定的区位码如表3-4所示。

例如，汉字“啊”的区位码为1601。它在机内的表示形式为：7F 2E 1D。在BASIC程序中用字符串函数表示则为：

A $＝CHR $(127)＋CHR $(46)＋CHR $(29)

3.6 显示控制操作

3·6·1 设备控制字

汉字系统支持的输出设备包括显示器、打印机和串行通信口。汉字系统通过使用设备控制字963单元($3C3) 来控制设备的工作状态以及确定控制字符和机内码的使用方法。设备控制字的定义如图3-22。

系统初始化时，963单元设置为 $01。即第0位为1，其它各位都为0。

第6位为0时，系统对控制字符作正常处理。第6位为1时，系统对控制字符不作正常处理，而是显示或打印该控制字符所对应的图形符号。

第7位为0时，汉字内码为学习机内码，即7F、区码、位

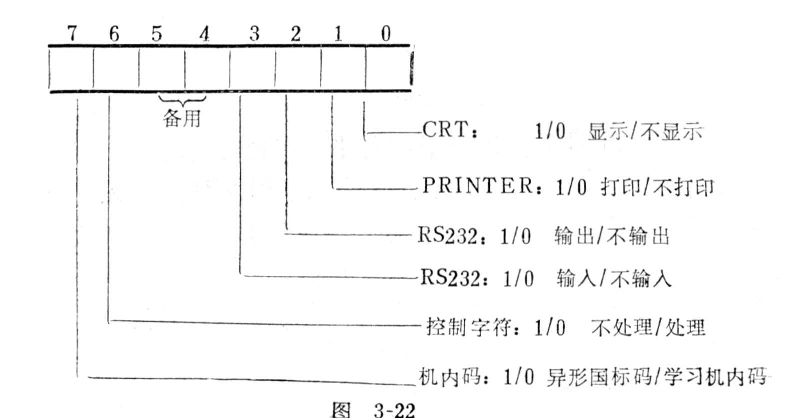


图 3-22

码。第7位为1时，汉字内码为异形国标码，即异形国标码高字节和异形国标码低字节。

通过设置设备控制字的值，可以实现打印和显示同时进行，也可以实现只打印不显示，以及串行通信的收、发操作。详细的情况见相应章节，即汉字打印的控制及串行通信的操作。

3.6.2 页面切换

进入汉字系统后，用户可以使用6个显示页面。页面1至页面4是高分辨率图形方式。页面5和页面6是倍高分辨率图形方式。页面1对应主存的 \$2000～\$3FFF，页面2对应主存的 \$4000～\$5FFF，页面3对应辅存的 \$2000～\$3FFF，页面4对应辅存的 \$4000～\$5FFF。页面5是同时使用页面1和页面3的存储器空间。页面6是同时使用页面2和页面4的空间。如图3-23所示。

初进汉字系统时，系统使用第2页。

页面的切换可以有两种方式，即键盘方式和程序方式。键盘方式为敲入Tab键后紧接着敲入页面号。例如，敲入Tab键和6键，屏幕变为第6页，进入倍高分辨率显示方式。程序方式切换页面可以通过输出一个控制码Ctrl-I和一个

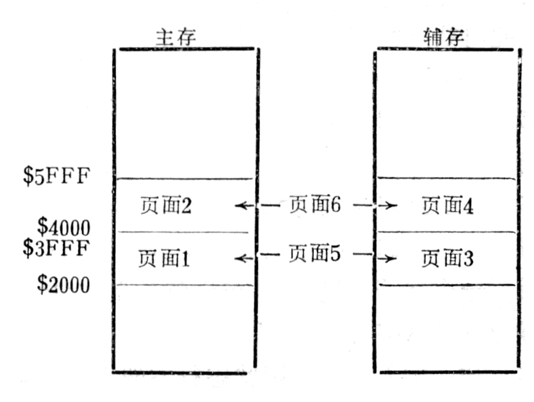


图 3-23

页面号来实现。例如：

```
10  PRINT CHR$ (9) ; "6"
```

此语句可使屏幕进入第6页。

在高分辨率方式下，每行可显示34个字符或17个汉字。在倍高分辨率方式下，每行可显示68个字符或34个汉字。

若用户使用辅存的页面显示，则主存空间可以空出来放用户程序。在高分辨率图形方式下可以从 $800用到 $9600，即BASIC程序有35K多的空间可以使用。

在退出汉字系统时应通过F6键，或输出Ctrl-Q。若直接在汉字方式下打入PR#6引导DOS操作系统，则应保证显示页面在主存空间。即在页面1或2。若在页面3至页面6，则引导DOS时会使屏幕变花。即屏幕仍显示辅存信息，而监

控却在使用主存显示。

3.6.3 状态行

状态行可以通过敲入Ctrl-R控制。敲一次Ctrl-R，关闭状态行，此时状态行可以作为屏幕显示区的最底行使用。再敲一次Ctrl-R，打开状态行，此时作为显示区的最底行自动向上移动一行，即整个屏幕向上翻滚一行，状态行又出现在最底行。

在状态行处于打开状态时，除左端有字符两字外，右端有键入字符的提示信息。提示信息总是显示当前内存中键入缓冲区的最后一个字符或汉字。对于字符信息，左边有“*”号表示字符为可显示字符，左边有“∧”号，表示字符为控制字符。当回车后，键入区空，则状态行右端没有任何信息。只有在字符输入方式下，状态行右端才有键入区提示信息。有了这个信息，在退格删除键入区中字符时十分有用。尤其是在使用多字符组的情况下，可以准确地删除字符和字符组的标志信息。在倍高分辨率方式下，此信息在状态行的中间。

3.6.4 12行汉字显示

初进汉字系统时，屏幕为11行显示。第11行可以是状态行也可以是屏幕显示行，用户可以通过Ctrl-R来选择。当执行下面语句时，屏幕变为满屏12行。即

```
10  POKE  964, 128
20  POKE  35, 11
```

若希望改回11行可执行下述语句：

```
10  POKE  964, 0
20  POKE  35, 10
```

3.6.5 屏幕开窗口

汉字系统下的屏幕可以开窗口，即可以将显示屏幕缩小并设定在任意位置。开窗口可以通过修改屏幕的边界值来实现。屏幕边界值单元使用如下：

左边界：$20（32）

宽度：　$21（33）

上边界：$22（34）

下边界：$23（35）

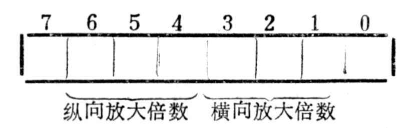
例如，下述程序将屏幕显示改为在屏幕中间的一个小窗口。

```
10  POKE  32, 10
20  POKE  33, 20
30  POKE  34, 3
40  POKE  35, 8
```

3.6.6 放大显示及彩色

在汉字系统下可以放大显示汉字和字符，并可以对放大的汉字及字符加上颜色。

放大控制单元为965（$3C5）。单元内的定义如下：



此单元初值为0，即对任何信息都不放大。一个汉字是16×16点阵的图形，最大可以放到128×128点阵的汉字及128×64点阵的西文字符。即横向和纵向都是最大放大到原

来点阵的8倍。例如，

```
10 POKE 965, 63
20 PRINT "中华"
```

上述程序使显示放到最大。

只有放大后的信息才能出现彩色。彩色的设置分两种情况。在高分辨率显示方式下，可以用下面的语句设置颜色。

```
HCOLOR=X
```

其中X取值为0～7。

另一种情况是在倍高分辨率显示方式下，设置颜色可用下面的语句：

```
COLOR=Y
```

其中Y取值为0～15。

在显示彩色放大的汉字时，显示效果受到所选择的颜色和放大倍数的影响。尤其是在倍高分辨率显示方式下，横向放大倍数应大于3倍，否则字型会受到较大影响。如果选择白色，即COLOR=15，则不受放大倍数的影响。

下面举两个例子。第一例程序将在高分辨率显示方式下显示出各种颜色的字符。程序如下：

```
10 POKE 965, 15
20 FOR I=0 TO 7
30 HCOLOR=I
40 PRINT"中";
50 NEXT
```

第二例程序将在倍高分辨率显示方式下显示出所有颜色的字符。程序如下：

```
10 POKE 965, 15
20 FOR I=1 TO 16
30 COLOR=I
```

```
40  PRINP"华";
50  NEXT
```

彩色的控制有两个层次。底层是硬件的彩色控制开关，位于机壳背后。见第二章中的有关介绍。上层是彩色控制软开关。显示出彩色的条件是硬件控制开关位于允许彩色显示位置，软开关处于彩色接通状态。第一次进入汉字系统时，汉字系统将彩色控制软开关切断，以后只要不作冷启动，再次进入汉字系统时保留彩色控制软开关的状态。但要注意，Ctrl-Reset将使软开关处于允许彩色显示状态。

彩色控制软开关的定义如下：

49245（$C05D）　开彩色

49244（$C05C）　关彩色

执行POKE 49245，0 可以开彩色，而执行POKE 49244，0 可以关彩色。

3.6.7 反相显示

文本区和状态行的反相显示是相互独立的。文本区的反相控制单元为主存储器的零页单元50（$32），而状态行的反相控制单元为辅存储器的802（$322）单元。执行下述语句可以改变文本区显示方式：

POKE 50，0　　反相显示

POKE 50，255　　正相显示

对于状态行显示方式可以执行下述语句来改变，反相显示为：

```
10  POKE 49157，0
20  POKE 802，0
30  POKE 49156，0
```

正向显示为：

```
10  POKE 49157，0
20  POKE 802，255
30  POKE 49156，0
```

3.6.8 屏幕编辑命令

屏幕编辑功能和监控程序所提供的完全相同。通过键盘键入的有关命令可以移动光标所在位置，而不改变键盘输入缓冲区的内容。这些编辑命令以“Esc”键为第一键，以一个字母键为第二键。若第二键为I，J，K和M，则光标移动后仍处于编辑状态，即I，J，K和M可连续使用，直到按下一个非编辑功能键。若第二键为A，B，C，D，E，F，@则光标移动后便自动退出编辑状态。下面列出所有编辑命令：

Esc A　光标右移一格，退出Esc状态

Esc B　光标左移一格，退出Esc状态

Esc C　光标下移一格，退出Esc状态

Esc D　光标上移一格，退出Esc状态

Esc E　从光标清至行尾，退出Esc状态

Esc F　从光标清至页尾，退出Esc状态

Esc @　清屏幕，光标回（0，0），退出Esc状态

Esc I　光标上移一格，仍保持Esc状态

Esc J　光标左移一格，仍保持Esc状态

Esc K　光标右移一格，仍保持Esc状态

Esc M　光标下移一格，仍保持Esc状态

3.6.9 输出字符控制命令

中华学习机除了提供上述键盘方式的屏幕编辑命令外，还提供了一些程序方式的屏幕编辑命令和输出字符控制命

令。它们的功能见表3-5。

例如，当在程序中希望清屏幕时，可用下面语句：

```
10 PRINT CHR$ (12)
```

表 3-5

DEC	HEX	命令功能
CHR$ (7)	$07	扬声器“嘟”叫一声
CHR$ (8)	$08	光标退一格
CHR$ (11)	$0B	从光标清至页尾
CHR$ (12)	$0C	清屏幕，光标为（0，0）
CHR$ (13)	$0D	输出回车符（CR），光标移至下一行
CHR$ (14)	$0E	置显示方式为正常方式
CHR$ (15)	$0F	置显示方式为反相方式
CHR$ (17)	$11	退出汉字系统
CHR$ (18)	$12	显示或清除状态提示字符
CHR$ (19)	$13	暂停输出，按任一键恢复输出
CHR$ (26)	$1A	从光标清至行尾

3.7 打印控制操作

在汉字系统下可以进行汉字的打印输出。汉字打印驱动程序是针对MX-80Ⅲ型打印机设计的，它要求打印机的接口为Centronics接口。对于兼容EPSON MX-80Ⅲ型的打印机，如FX-80Ⅲ、RX-80Ⅲ、CP-80Ⅲ、FX-100+和YAMATO等型号也可以使用。对于MX-80Ⅱ、FX-80Ⅱ、RX-80Ⅱ等型号的打印机除压缩的字型不能打印外，其他的汉字也都能打印出来。

3.7.1 打印机的启动

启动打印机有两种方式，一种是键盘方式，另一种是程序方式。键盘方式是敲入Ctrl-O，于是显示器和打印机同时输出。再按一次Ctrl-O，打印机停止输出。

程序方式启动打印机可以通过对设备控制字的设置来实现。若要显示和打印同时输出，可以用

POKE 963，3

若要只打印不显示可以用

POKE 963，2

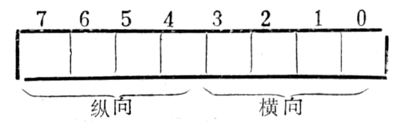
停止打印输出可以用

POKE 963，1

963单元的定义可参见设备控制字的定义一节。

3.7.2 横向放大和纵向放大

打印汉字的字型可以通过修改横向放大和纵向放大的参数来确定。这个参数单元就是字形选择单元，参数值就是字形号。此单元的地址为1659（$67B）。单元内每一位的定义如下：



即低4位确定横向放大倍数，高4位确定纵向放大倍数，取值都是0～15。整个字节的取值为0～255。因此共有256种字型。不论是横向或纵向，基本单位是16个点。每放大一倍就增加16个点，因此横向或纵向最大可放大到256个点。但由于打印机在打印汉字时工作在倍密度图形打印方式下，横向

每打一点打印头只移动半个点的位置，有半个点与前面的点重合，因此在横向放大倍数与纵向放大倍数相等时，虽然打印点数在两个方向上是相同的，但打印出的汉字其横向距离大约是纵向距离的二分之一，即字是长方形的。例如下面这个程序打印出横向纵向各放大15倍的汉字，也就是最大的放大字型。

```
10  POKE 1659，255
20  POKE 963，3
30  PRINT"学习"
40  POKE 963，1
```

打印结果如图3-24。

图 3-24

当字号取0时，也就是横向纵向都不放大时，可以打印出压缩的字型。当然这要求打印机在纵向走纸时有走半个点的功能。通常Ⅲ型打印机都有这种功能而Ⅱ型则没有。下面程序可以打出压缩的字型。

```
10  POKE 1659, 0
20  POKE 963, 3
30  PRINT"中华学习机CEC-IA型"
40  POKE 963, 1
```

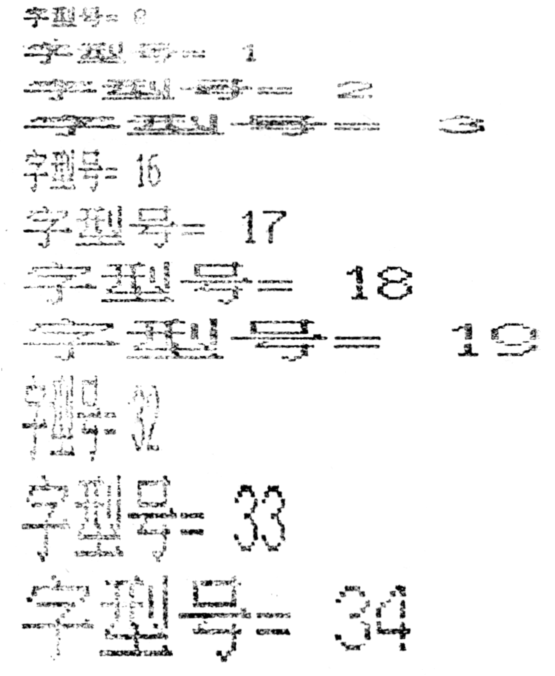


图 3-25 几种

打印结果为

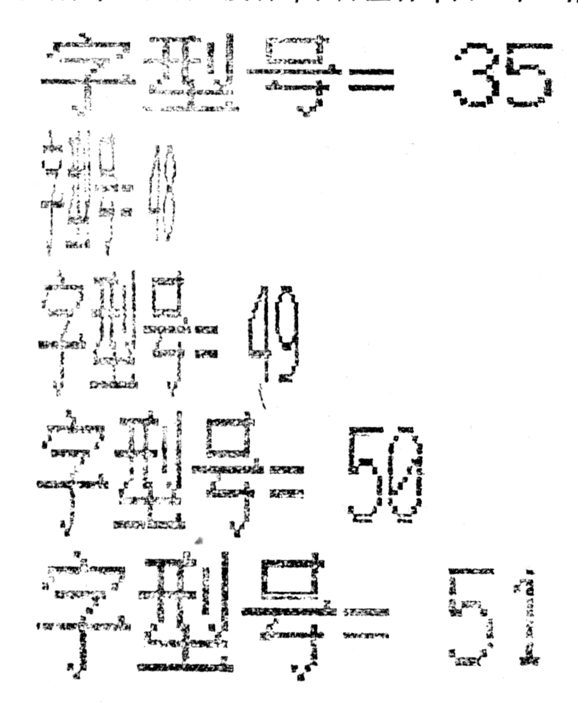
中华学习机 CEC-IA型

为了使Ⅲ型和Ⅱ型都可以打印，汉字系统初始化时将字型设置为横向纵向各放大1倍。即1659的初值为17($11)。

图3-25是几种常用的字号和字型。

3.7.3 下划线、旋转90度和加重打印

下划线打印、旋转90度打印和加重打印由一个工作单元



常用的字号和字型

控制，地址为962（$3C2）。各位的定义如图3-26。

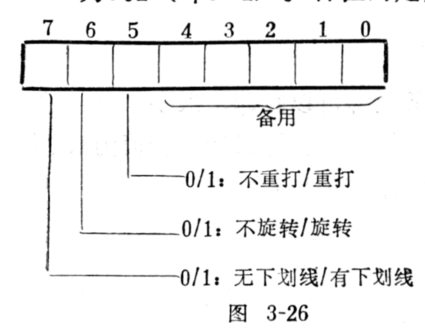


图 3-26

下划线打印的例子如下：

```
10  POKE 962，128
20  POKE 963，3
30  PRINT"计算机"
40  POKE 963，1
50  POKE 962，0
```

运行结果如下：

计算机

旋转打印的例子如下：

```
10  POKE 96 ，64
20  POKE 963，3
30  PRINT"白日依山近"
40  PRINT"黄河入海流"
50  PRINT"欲穷千里目"
60  PRINT"更上一层楼"
70  POKE 963，1
80  POKE 962，0
```

运行结果如图3-27。

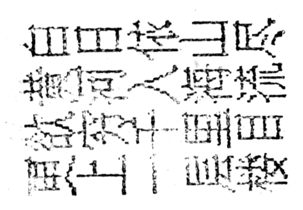


图 3-27

加重打印的例子如图3-28。

```
10  POKE 963, 3
20  PRINT "物质是由"
30  POKE 962, 32
40  PRINT "分子"
50  POKE 962, 0
60  PRINT "组成"
70  POKE 963, 1
```

运行结果如图3-29

物质是由分子组成

图 3-29

注意其中分子两字是加重打印的。加重打印在色带较浅时效果较明显。

3.7.4 间距和纸宽

在打印汉字时，可以控制字间距，行间距及纸宽。

字间距的控制单元为1787（$6FB），初始值为1，即字与字之间空一个点。允许字间距在0～255之间。

行间距的控制单元为1915（$77B)，初始值为1，即行与行之间空一个点。允许行间距设置在0～72之间。

一个点位置相当于1/72英寸。

纸宽的控制单元为2043（$7FB），初始值为80，即字符打印方式下的80个西文字符宽。通常打印机上有标尺，其刻度就是以字符打印方式下西文字符为单位。当纸宽确定后，不论选择什么字型，当一行信息在纸宽以内打不下时自动换行打在下一行。这样可以不必计算每行允许的打印字符个数，也不会因为超过纸宽而丢失信息。

3.7.5 多字符组和反相打印

多字符组的组号存放在966（$3C6)单元。字符组的规定可参阅显示控制操作的相应章节。通常，在输入操作时已将字符组的转换控制字符Ctrl-F和字符组号输入了字符串，因此打印输出时不必作特殊操作，只要启动打印机即可打印出多字符组。

若直接由程序控制输出打印各种字符组的字符，可以通过修改966单元实现，即将所选字符组号填入966单元即可。

反相打印控制单元为50（$32），和反相显示控制单元是同一个单元。填入255是正向打印，填入0是反相打印。

第四章　CEC-BASIC语言的使用方法

在CEC-ⅠA主机板上的只读存储器内固化了CEC-BASIC解释程序，从而使用户一开机就可以方便地使用BASIC语言。

CEC-BASIC具有全部APPLESOFT BASIC语言的功能。此外CEC-BASIC独具以下特点：

一、可在西文40列、西文80列、汉字17列及汉字34列等显示方式下使用。

二、比APPLESOFT BASIC多增加了5条指令：

MUSIC、PLAY、SASM、BCLR、MNT

三、比APPLESOFT BASIC多增加了二种函数：

DEC、HEX$

四、增强了APPLESOFT BASIC读写磁带语句LOAD和SAVE的功能与可靠性。

4.1　CEC-BASIC的基本规定

4.1.1　常数与变量

CEC-BASIC允许使用正、负实型常数，数值的范围是$2^{-128} \leqslant |x| \leqslant 2^{127}$，即

$2.9387 \times 10^{-39} \leqslant |x| \leqslant 1.7014 \times 10^{+38}$

当BASIC解释程序遇到绝对值小于2.9387×10^{-89}时就作为0处理，大于$1.7014\times10^{+88}$时出现溢出错误。

CEC-BASIC的有效实型数据位为9位，超过9位用舍入处理。在输出实型数时，若

$$0.01\leqslant|x|\leqslant999999999.2$$

则x用定点表示法输出。若超过此范围则用科学计数法表示。格式如下：

Sn.nnnnnnnnE±TT

其中S表示该数的符号，如果符号为“+”时则被省略，n是0～9的数字，TT为指数值，用两位数表示。例如$-12.34567896\times10^{10}$在BASIC里表示为-1.23456789E+11。

BASIC里允许出现正、负两类整型常数。整型数值的范围是$|x|\leqslant2^{15}$，即

$$-32767\leqslant x\leqslant+32767$$

CEC-BASIC规定字符串常数用双引号"括起来，引号内可以是字符、数字、符号及汉字等。

CEC-BASIC语言内除常数外，还可以使用变量。为区分不同的变量，用户需给变量起名字。在给变量定名时要注意如下这些规定：

一、变量名的第一个字符必须是字母字符，字母之后可以是数字或字母。例如，XY2、A、B、A1、B2等都可以作为变量名使用。变量名最多不得超过238个字符。

二、BASIC解释程序只识别变量名最前面的两个字符，例如WUGD和WUANG表示同一个变量。

三、不能用BASIC使用的语句、命令和函数等作为变量

名。

同常数一样变量也有三种类型：实型变量、整型变量和字符串变量。实型变量直接用变量名表示，整型变量用变量名紧接%来表示，字符串变量用变量名紧接$表示。这三种变量可由表4-1概述：

表4-1

变量类型	范　　围	表示方法	例
实　型	-38≤指数≤+38　尾数：9位	变量名	A,BC,D1,CITY
整　型	±32767之间	变量名%	I%，JK%
串　型	小于255个字符	变量名$	A1$，AR$

实型和整型之间可以进行转换，这是由BASIC解释程序自动完成的。实型变量赋给整型变量时，取不大于该数的整数，这与取整函数INT的效果一样。如果整型变量赋给实型变量，其值不变。例如：

```
LET X=5.98
LET A%=X
PRINT A%
```

显示结果

```
5
```

这说明整型变量A%已赋值5，即5.98的的整数部分。

但字符串类型的变量与实型或整型变量转换时要借助于BASIC的函数完成。

数值变量在未赋值前，其值为零；串变量未赋值前为空，即没有任何字符。

4.1.2 运算符与表达式

CEC-BASIC中三类运算符：

第一类是算术运算符，它们是：

+（加），-（减），*（乘），/（除），∧（乘方）

第二类是逻辑运算符，它们是：

<>（不等于），><（不等于），

=（等于），>（大于）、<（小于），

>=（大于等于），=>（大于等于），

<=（小于等于），=<（小于等于），

AND（与），OR（或），NOT（非）。

第三类是串运算符，它只有一个：

+（串连接）

用运算符将数据、变量或函数连接成的运算式称为表达式。

表达式分三种类型：

第一类是算术表达式，它是用算术运算符将整数、实数、算术变量、算术函数连接起来的运算式。此外，单独形式的整数、实数、算术变量、函数也作为算术表达式的特殊形式。下面这些运算式都是算术表达式：

2*3，1.5∧(1/3)，3.1415926，3+COS(0.5)

第二类是串表达式，这种表达式是用串运算符将字符串、串变量、串函数连接起来的运算式。单独形式的串变量、字符串也作为串表达式的特殊形式。例如：

A$+B$，"CEC-ⅠA"，"ABC"+"123"

第三类是逻辑表达式，这种表达式是用逻辑运算符将任

何种表达式连接起来的运算式。逻辑表达式的运算结果为逻辑值，逻辑值只能为0或1，0表示“假”，1表示“真”。下面这些表达式是逻辑表达式：

NOT (5>6) ，A>0，B< >3

(A$＝″NO″) OR (B$＝″YES″)

一个表达式中可以包含多种运算符，BASIC解释程序在处理表达式时根据运算符的优先级别决定求值的顺序，当有括号（ ）出现时先处理最内层括号中的运算式，在运算式中，各运算符的优先次序如下：

1. +，-，NOT
2. ∧
3. *，/
4. +，-
5. >，<，>=，=>，<=，=<，<>，><，=
6. AND
7. OR

表达式中相同优先级的运算符的运算次序是从左到右。

4.1.3 保留字、语句行的规定及执行方式

CEC-BASIC使用了一些英文单词或英文单词的缩写作为BASIC语言本身的命令、语句和函数名称，这些词或词的缩写称为保留字，例如IF、THEN、PRINT、INPUT、HEX$等。保留字是不允许作为变量名使用的。例如下面这些用法都是错误的：

A＝INPUT，HEX$＝″CEC″

IF X＝A THEN PRINT″X＝A″

最后一例出错是因为X = A的字母A和下一个字母T合为AT是BASIC的保留字，这类错误应格外当心。附录D中给出了BASIC的全部保留字。

CEC-BASIC对语句行有一定规则，具体内容如下：

1. 一个语句行必须以行号开始，以回车键结束。

2. 语句行号必须是无符号的整数，其范围是0～63999之间。

3. 一个语句行中可以包括多个语句，语句和语句之间必须用冒号（:）分隔。但每一个语句行中最多只可输入239个字符。

关于BASIC语言的执行方式共有两种。第一种是立即执行方式。当进入BASIC语言状态时，屏幕上显示出“]”和一个光标（40列西文下是花格的闪烁方块，在80列西文下是不闪烁的实心方块）。此时不键入语句行号，而直接键入BASIC的语句或命令，按回车后，这个语句或命令会立即被执行，这种执行方式被称为立即执行方式。

另一种方式称为暂缓方式，或称作编程方式。当屏幕上出现]和光标后，键入一个行号（0～63999中的一个整数），接下去继续键入BASIC的语句或命令，再键入回车，则这一行语句就被存到计算机的存储器中，并继续等待输入新的语句行。当所有的语句都输入完后，可用RUN命令执行刚输入进去的那些语句。

由BASIC语句行组成的语句集合称为BASIC程序。程序一般从最小行号的那一行开始执行，如果无转移，则按顺序从小行号到大行号逐行执行。

4.2 CEC-BASIC的语句

4.2.1 描述语句与函数的特殊符号

为了严格而简明地描述BASIC语句或函数的格式，本书使用如下三种特定符号对语句或函数格式进行说明：

1. 符号|——分开任选其中之一的各项，例如a|b表示选择a、或选择b。

2. 符号[]——括号内的内容可以省略，例如[a]表示a项可以要也可以不要。

3. 符号{}——括号内的内容可以重复，例如{a}表示可有一个a，也可以有多个a。

上述三种符号仅为描述格式用，用户在键入语句或函数时不要把它们输入进去。

4.2.2 赋值与输入输出语句

利用BASIC程序对数据进行处理或运算时，先要把数据交给计算机，计算机计算出结果后通过输出设备把结果告诉给用户。为此，BASIC语言提供了一类提供数据和输出结果的语句。

1. LET

格式：[LET]算术变量[下标]=算术表达式

[LET]串变量[下标]=串表达式

功能：将赋值号（=）右边的表达式的值赋给左边的变量。

方式：立即型与暂缓型。

例 1

```
LET A =3+4
B=3*4
```

例 2

```
10  PI=3.1415926
20  A (1) =PI/2
30  C$="CEC-IA"
40  CB$=C$+"BASIC"
```

在变量未赋值以前，算术变量为0，串变量为空，一旦被赋值则一直保留到重新被赋值为止。此外，表达式的类型必须与被赋值变量的类型一致。

2. DATA

格式：DATA[数据][{，数据}]

功能：存储数据，以供READ语句使用。

方式：暂缓型。

使用DATA时要注意数据之间用逗号分隔，串数据可以不用引号，但如果串中出现逗号，则必须用引号将字串括起来。例如：

```
10 DATA X,2.5,"CEC,BASIC",BEIJING
```

3. READ

格式：READ[变量名][{，变量名}]

功能：将DATA语句中的数据依次读到READ后面的变量中。

方式：立即型与暂缓型

例 1

```
10  READ X, Y, Z$
20  DATA 2, 5, CHINA
```

此例运行后，相当于执行了赋值语句LET，即

```
LET X=2
LET Y=5
LET Z$="CHINA"
```

使用READ语句时要注意变量的类型要与数据的类型一致，否则将出现句法错误。

例 2

```
10  READ X
20  DATA HELLO
]  RUN
? SYNTAX ERROR IN 10
```

另外，DATA语句中的数据应不少于READ语句中变量的数目，否则将出现超出数据错误。

例 3

```
10  READ X, Y, Z
20  DATA 1.5, 2
]  RUN
?  OUT OF DATA ERROR IN 10
```

4. PRINT

格式：PRINT[{表达式}[，|；[{表达式}]]][，|；]

功能：将各种数据或表达式的值输出到屏幕或指定的输出设备（如打印机、通信接口等）上。

方式：立即型与暂缓型。

例 1

```
] PRINT 2/3*2.25+0.01
```

```
]1.51
```

在CEC-BASIC里可用问号“？”代替输入“PRINT”语句。

例 2

```
] ?"CEC-BASIC PRINT"
] CEC-BASIC PRINT
```

PRINT中的表达式与表达式之间可用分号“；”或逗号“，”分隔。如果用分号作分隔符则前后表达式紧接着输出。

例 3

```
] PRINT"2+3=";2+3
] 2+3=5
```

如果用逗号作分隔符，则每个表达式占16个字符的位置。如果表达式的值超过15个字符，则下一个表达式后移16个字符位输出。如果表达式不能在同一行输出完则继续从下一行的起始端输出。

例 4

```
PRINT 5*7, 1, "W"
35              1               W
↑               ↑               ↑
第1列           第17列          第33列
```

例 5

```
PRINT 5*7, 0, 3.1415926, "W"
35              0               ← 第1行
3.1415926       W               ← 第2行
↑               ↑
第1列           第17列
```

因为第三个输出项无法在第33列到第40列（屏幕仅40列）之

间输出完，所以转到下一行的第一列输出。

在PRINT的最后一个表达式末尾出现分号或逗号时，表达式与表达式之间出现这两个符号的处理结果一样；如果不用分号或逗号作结束符，则下一次输出另起一行。

例 6

```
10  PRINT "A"
20  PRINT "B"
30  PRINT "C"
]  RUN
A
B
C
```

5. INPUT

格式：INPUT[“字符串”；] 变量[{，变量}]

功能：从当前的输入设备上（如键盘）接收各种类型的数据，依次赋给其后的变量。

方式：暂缓型。

当BASIC程序执行到INPUT语句时，屏幕上出现问号“？”和光标，等待用户从键盘上输入数据或字符，如果有多个数据要一次输入，可用逗号作分隔符，输入完毕后用回车键结束输入。

例

```
10  INPUT A, B, C
20  PRINT A+B+C
RUN
?  1, 2, 3
6
```

如果在输入时希望有提示，可在INPUT语句之后加入

提示的字符串，并用分号同后面的变量分开。如果有提示的字符串，则执行时问号会被字符串所代替。

例 2

```
10 INPUT"HOW OLD ARE YOU? "; X
20 PRINT"YOU ARE"; X
RUN
HOW OLD ARE YOU? 13
YOU ARE 13
```

当输入的数据少于变量个数时，显示双问号？？，等待继续输入。如果输入的数据多于要求输入的变量个数，则出现警告性错误EXTRAIGNOREO，程序仍继续执行，多输入的数据被忽略。

例 3

```
10 INPUT A
20 PRINT A*A
] RUN
? 4, 7
EXTRAIGNRED
16.
```

若要中断INPUT语句，应在输入第一个数据之前键入Ctrl-C，然后按回车键。

6. GET

格式：GET变量

功能：从当前的输入设备上接收一个字符或一位数字赋给变量。

方式：暂缓型。

例 1

```
10  GET A
20  PRINT A
]RUN
5
```

使用GET语句时，注意输入的数据要与变量类型一致。BASIC在执行GET语句时不在屏幕上显示输入的字符，只有通过输出语句PRINT才能显示该字符。此外GET语句只接收一个字符，且接收到后立即继续执行下面的语句。

7. RESTORE

格式：RESTORE

功能：将数据指针恢复指向第一个DATA语句的第一个数据。

方式：立即型与暂缓型。

当READ读了DATA语句里的一个数据后，便把数据指针指向下一个数据，以便下次继续读入数据。RESTORE语句的作用就是把数据指针重新指向第一个DATA语句里的第一个数据。

例

```
10  READ X, Y
20  PRINT X, Y
30  RESTORE
40  READ A, B
50  ?  A, B
60  DATA  1, 2, 3, 4
] RUN
1               2
1               2
```

8. PR#

格式：PR# 表达式

功能：选择输出设备。

方式：立即型与暂缓型。

语句PR#的作用是根据表达式的值选择输出设备的通道。

例

PR#0 选择显示器输出

PR#1 选择打印机输出

PR#2 选择通信接口输出

PR#6 选择磁盘驱动接口输出

9. IN#

格式：IN# 表达式

功能：选择输入设备。

方式：立即型与暂缓型。

CEC-ⅠA共有8个输入输出槽口，因此PR#和IN#中的表达式的值只能在0～7这个范围内取。

4.2.3 和执行顺序有关的语句

使用本节中的BASIC语句可以改变程序的执行顺序，使程序按用户的要求实现分支、循环、调子程序等功能。

1. GOTO

格式：GOTO 行号

功能：使程序转到行号指定的那一行语句上执行。

方式：立即型与暂缓型。

例

```
10  X=0
20  X=X+1
```

```
30  PRINT X
40  GOTO 20
```

此例是输出自然数1，2，3，…的程序，中断此程序按Ctrl-C或Ctrl-Reset键。

2. IF…THEN

格式：①IF　表达式　THEN 语句[{：语句}]

　　　②IF　表达式　THEN 行号

　　　③IF　表达式　GOTO 行号

功能：根据表达式值确定程序的转移方向。

方式：立即型与暂缓型。

如果表达式的值为0，表示条件不成立，程序转到下一条语句执行。如果表达式的值不为0，遇②、③两种格式时，程序转到指定行号的那条语句执行；若遇第①种格式，则执行THEN后面的那个（或那些）语句。

例 1　第①格式的情况：

```
10  INPUT"X=？"；X
20  IF X=0 THEN PRINT"X=0,PLEASE AGAIN":GOTO 10
30  PRINT"1/X="；1/X
```

此例是由用户从键盘输入一个数，当该数为0时，计算机显示"X=0，PLEASE AGAIN"，即让用户重新输入；当X不为0时，计算出该数的倒数，并显示。请运行之：

```
RUN
X=？0
X=0，PLEASE AGAIN
X=？2
1/X=0.5
```

例2 第②种格式：

```
10 READ Y
20 IF Y<=0 THEN 10
30 PRINT Y, "IT'S>0"
40 DATA 0, -1.5, -2, 1.7, 2
RUN
1.7  IT'S>0
```

此例先由READ从DATA中逐个读数据，当读到的数据小于或等于零时，继续往下读数据，即转到10行执行；当读到大于零的数据就显示它。

第③种格式与第②种格式的用法相同，用户可试将例2中的第20行换成

```
20 IF Y<=0 GOTO 10
```

其余部分不变，再运行例 2，其结果相同。

3．FOR…NEXT

格式：FOR 实型算术变量=算术表达式1
　　　TO算术表达式2[STEP算术表达式3]
　　　……
　　　……
　　　NEXT[算术变量][{，算术变量}]

功能：循环执行FOR与NEXT之间的程序。

方式：立即型与暂缓型。

FOR，NEXT语句亦称为循环语句，是BASIC语言中的非常有用的语句，先看一个例子。

例 1

```
10 FOR I=1 TO 10 STEP 1
20 PRINT I; ","; 
30 NEXT I
```

```
RUN
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

从此例可以看出，每执行一遍FOR和NEXT之间的语句（这些语句亦称为循环体），实型算术变量（亦称为循环变量，即例1中的I）便在算术表达式1（亦称循环变量的初值，即例1中的1）的基础上增加算术表达式3（亦称步长，即例1中的STEP 1）所给定的值。当循环变量超过算术表达式2（亦称循环变量的终值，即例1中的10），则循环体不再执行。

如果步长为1，则通常省略STEP 1。其余情况均不能省略STEP。NEXT后的变量也可省略。

例 2

```
10 FOR I=0 TO 10 STEP 2
20 PRINT I; ", "; : NEXT
RUN
0, 2, 4, 6, 8, 10
```

在循环体内还可以再出现FOR_NEXT语句，此情形称为循环嵌套。

例 3

```
10 FOR I=1 TO 4              ┐
20 FOR J=1 TO 4  ┐第二层     │第一层
30 PRINT"*"; : NEXT ┘        │
40 PRINT: NEXT               ┘
RUN
* * * *
* * * *
* * * *
* * * *
```

4. GOSUB RETURN

格式：GOSUB行号

…

RETURN

功能：执行GOSUB语句时，转到行号指定的BASIC语句执行，遇RETURN语句时返回到GOSUB的下一行继续执行。

方式：立即型与暂缓型。

GOSUB语句亦称子程序调用语句，用户在编程时，可能有某些语句要多次重复执行，用子程序可以缩短程序长度，还能使程序有条理。

例 1　计算（5！+3！）/7！

（n！= n×（n－1）×（n－2）×…×3×2×1）

```
10  READ A, B, C
20  DATA 5, 3, 7
30  N=A: GOSUB 100
40  A=S   (A中保存5！)
50  N=B: GOSUB 100
60  B=S   (B中保存3！)
70  N=C: GOSUB 100
80  C=S   (C中保存7！)
90  PRINT"(5! +3! )/7! =";
95  PRINT (A+B)/C
99  END
100  S=1          ┐
110  FOR I=1 TO N │
120   S=S*I       ├ 子程序部分
130  NEXT         │
140  RETURN       ┘
```

5. POP

格式：POP

功能：将堆栈中的地址弹出，但不返回最近执行的GO-SUB的后继语句，而直接执行POP的下一个语句。

方式：立即型与暂缓型。

在某些时候，用户的子程序并不希望返回到调用它的语句的下一个语句,而希望转到别的语句执行，POP可以实现这一点。

例 1 计算 $\frac{1}{2}+\frac{1}{7}+\frac{1}{9}$

```
10  READ X
20  GOSUB 100
30 GOTO 10
40  PRINT"IT IS", S
50  DATA 2, 7, 9, 0
60  END
100  IF X=0 THEN POP: GOTO 40
110  S=S+1/X
120  RETURN
```

6. ON GOTO

格式：ON 表达式 GOTO 行号1，行号2，…，行号i

功能：根据表达式的值（范围0～255）n，选择第n个行号语句去执行。

方式：暂缓型。

ON、GOTO语句是一种分支转移的语句，如果表达式的值为1，就转到第一个行号语句执行，为2就转到第 2 个行号语句执行。如果表达式的值超出0～255，则将出现错误；如果表达式的值为0或超出了GOTO所提供的行号，则转到

ON、GOTO语句的下一条语句执行。

下例是用户通过键入1、2、或 3 来选择显示三个地 名：北京、上海、广州。

```
10  PRINT"PLEASE CHOOSE (1-3) :";
20  GET X
30  ON X GOTO 100, 200, 300
40  GOTO 10
100  ?"BEIJING": END
200  ?"SHANGHAI": END
300  ?"GUANGZHOU": END
```

7. ON GOSUB

格式：ON 表达式 GOSUB 行号1、行号2、…，行号i

功能：根据表达式的值（范围0～255）n，选择第n个行号，使程序调用第n行的子程序。

方式：暂缓型。

ON、GOSUB与ON、GOTO的用法类似，区别仅在于ON GOTO直接转到行号语句执行而不再返回；而ON GOSUB则调用行号指向的子程序，从子程序返回后仍执行ON GOSUB的下一条语句。

8. ONERR GOTO

格式：ONERR GOTO行号

功能：程序出错后转由行号给定的出错处理程序。

方式：暂缓型。

当程序运行过程中，产生错误时，BASIC解释程序会中断程序的运行。如果用户不希望中断程序，而希望自己把错误消除后继续运行程序，可用本条语句。当发生错误时，BASIC解释程序在内存222（$DE）中放入错误类型码。类

型码的意义如下：

类型码	出错情况
0	NEXT与FOR不匹配
16	句法错
22	RETURN与GOSUB不匹配
42	缺少数据
53	非法量
69	溢出
77	缺少内存
92	未定义语句
107	不适当下标
120	重定义数组
133	除数为零
163	类型不匹配
176	串太长
191	形式太复杂
224	未定义函数
254	响应INPUT时输入不适当
255	用Ctrl-C中断程序

在显示器上显示的英文句子分别是：

? NEX T without FOR error

? Syntax error

? RETURN without GOSUB error

? Out of data error

? Illegal quantity error

? Overflow error

? Out of memory error

? def'd statement error

? Bad subscript error

? Redim'd array error

```
? Division by zero error
? Type mismatch error
? String too long error
? Formula too complex error
? Undef'd function error
? EXTRA IGNORED
Break in 120
```

一般222单元的内容可用函数PEEK（详细用法见函数一节）读到。下例是不停地在屏幕上显示A字符，当你按下Ctil-C时，程序就断下来，并显示“STOP！”

例 1

```
10  ONERR GOTO 30
20  PRINT"A"; : GOTO 20
30  IF PEEK(222)=255 THEN? "STOP! "
RUN
AAAAAAAAAAAAAAA
STOP!
```

使用ONERR GOTO语句时注意应把ONERR GOTO放在出错误的语句之前，通常放在第一条语句。

9. RESUME

格式：RESUME

功能：返回发生错误的那条语句继续执行。

方式：暂缓型

RESUME的作用是从出错处理程序返回到发生错误的那条语句继续执行。

例 1

```
10  ONERR GOTO 30
20  PRINT X; ", "; : GOTO 20
```

```
30  X=X+1
40  RESUME
```

此程序运行后，先在屏幕上出现0，0，…，0，当按下Ctrl-C后显示的值增加一个。若要中止此程序必须按Ctrl-Reset键。

4.2.4 和系统有关的语句

1. SAVE

格式：SAVE[“文件名”]

功能：将BASIC程序存入磁带。

方式：立即型与暂缓型。

在程序存入磁带时，先键入SAVE，如果想给程序起个名字再键入“名字”，再将录音机的PLAY和REC键按下（即使录音机处于录音状态），最后键入回车键。当屏幕上显示BASIC的提示符“]”后，表示程序录制完毕，即可按下录音机的停止键。

2. LOAD

格式：LOAD[“文件名”]

功能：将磁带上的BASIC文件读入内存。

方式：立即型与暂缓型。

当用户用SAVE命令将BASIC程序写入磁带后，可用LOAD将其重新调入内存。方法是，先将磁带退到存入程序的起始位置（可根据录音机的计数器定位），再键入LOAD及“文件名”（文件名要与SAVE的一致）并立即按下录音机的PLAY键。当屏幕上再次出现BASIC的提示符“]”时程序已装入内存。

在使用SAVE或LOAD时都要注意文件名可以不要，如果要了文件名必须用引号括起来，否则将与DOS的SAVE和LOAD混淆。在SAVE程序时，对录音机的音量调节无特殊要求；在LOAD程序时应把音量调到适中的位置。如果LOAD之后很久没有把程序调入内存或出现“ERR”字样，表示程序未装入成功。可检查操作是否正确或修正音量，直至装入正确为止。

3. NEW

格式：NEW

功能：删除内存中的BASIC程序。

方式：立即型或暂缓型。

一般在输入新的BASIC程序前要先执行此语句，清除内存原有的BASIC程序。

4. RUN

格式：RUN

功能：运行BASIC程序。

方式：立即型与暂缓型。

5. STOP

格式：STOP

功能：中止程序的执行。

方式：立即型与暂缓型。

执行了STOP语句后，程序便停在这条语句上，并显示“BREAK IN 行号”，利用STOP语句，用户可检查和修改程序。如果想继续执行可键入CONT命令。

例 1

```
10  A=3
```

```
20  STOP
30  PRINT A
] RUN
BREAK IN 20
] PRINT A
3
] A=A+2
] CONT
5
]
```

6. CONT

格式：CONT

功能：如果程序是由STOP，END和Ctrl-C停止运行的，那么CONT可使程序从下一条语句开始执行。

方式：立即型与暂缓型。

使用CONT时应注意，如果程序停止后又增加或删除过程序的某些内容，则因程序在内存的断点地址发生了变动而不能再在原断点处继续执行。此时若执行CONT命令会出现错误信息：

```
? CAN'T CONTINUE ERROR
```

7. Ctrl-C

格式：Ctrl-C（同时按下Ctrl和C键或单按Break键）

功能：中断正在执行的BASIC程序。

方式：立即型。

当程序运行时按Ctrl-C，即可中断程序的运行，并显示

```
BREAK IN 行号
```

因此在调试程序或程序无法停下来（死循环）时，Ctrl-C命令是非常有用的。

8. END

格式：END

功能：终止程序的运行。

方式：立即型与暂缓型。

一般情况下在程序结束时使用此语句，如果END后面没有其它语句，也可以省略它。

例 1

```
10  PRINT"CEC-IA PROGRAM"
20  END
```

9. MNT

格式：MNT字符串表达式

功能：执行由字符串表达式所给的监控命令。

方式：立即型与暂缓型。

此语句是CEC-Ⅰ A新增加的BASIC语句。其中字符串可以是任何监控命令（关于监控命令的使用方法，请参考第五章）。

例 1

```
10  A$="300:4C 58 FC  "
20  MNT A$
30  MNT"300L N  300.303"
    RUN
```

在此例中，语句10将一个修改存储器内容的监控命令作为字符串赋到串变量A$中，语句20执行了这条监控命令。语句30列出了执行的结果，即内存地址$300开始的反汇编内容，然后又显示了$300到～$303的数据。读者不妨在机器上

运行此例看一看效果。

10. TRACE

格式：TRACE

功能：将所执行的BASIC程序的每一个行号显示出来。

方式：立即型与暂缓型。

TRACE的功能是跟踪程序，使用户能看到所运行的每一条语句的行号。

例 1

```
10  TRACE
20  FOR  I=1 TO 3
30  PRINT",”
40  NEST
]  RUN
#10  #20  #30,
#40  #30,
#40  #30,
#40
]
```

11. NOTRACE

格式：NOTRACE

功能：解除TRACE建立起来的跟踪功能。

方式：立即型与暂缓型。

12. POKE

格式：POKE 算术表达式 1，算术表达式 2

功能：把算术表达式 2 的值存到以算术表达式 1 为地址的内存中。

方式：立即型与暂缓型。

使用POKE时，注意算术表达式 1 的范围是－65535～65535，算术表达式 2 的范围是0～255。POKE语句会直接影响到系统内部使用的内存单元，通常POKE语句要在了解CEC－Ⅰ A内存分配的前提下才能使用，也可根据本书所提供的例子使用POKE语句。

例 1

```
10  FOR I=0 TO2
20  POKE 766, I
30  PR#3: PRINT
40  PRINT"CEC-IA"
50  GET A$
60  NEXT
```

766单元是显示方式单元。在766单元分别放入0、1 和 2，再执行PR#3和PRINT命令，可使显示方式分别在西文40列、汉字和80列方式下。例1即在以上三种显示方式下显示"CEC－Ⅰ A"的字样。

13. WAIT

格式：WAIT 表达式1，表达式2[，表达式3]

功能：使程序在执行过程中处于等待状态，直至条件被满足后才能继续执行。

方式：立即型与暂缓型。

为方便描述WAIT语句的用法，用X代表表达式 1，用{X}代表X指定内存地址中的内容，Y代表表达式 2；Z代表表达式 3。其中X，Y和Z的取值范围是

$$-65535 \leqslant X \leqslant +65535$$

$$0 \leqslant Y \leqslant 255, \quad 0 \leqslant Z \leqslant 255$$

{X}∧Y代表X所指的内存单元与Y按位进行"与"运算，例

如，设{768}=142，即二进制的10001110，Y=15即二进制的00001111，那么{768}∧15也即

```
    1 0 0 0 1 1 1 0    十进制142
∧   0 0 0 0 1 1 1 1    十进制 15
--------------------
    0 0 0 0 1 1 1 0    十进制 14
```

其结果便是14。与运算的规律是两位中任意一位为 0，则结果为 0。

（{X}∀Z）∧Y表示X所指的内存单元与Z值进行二进制的异或运算，其结果再和Y值进行与运算。两位二进制位异或的规律是当两位不相同时，其结果为 1。

如果WAIT语句以

WAIT X，Y

这种格式出现，则

{X} ∧Y = { 零，程序重复执行WAIT语句；非零，程序执行WAIT后面的语句 }

如果WAIT语句以

WAIT X，Z，Y

这种格式出现，则

（{X} ∀Z）∧Y = { 零，程序重复执行WAIT语句；非零，程序执行WAIT后面的语句 }

例 1

```
10 PRINT"PRESS ANY KEY TO CONTINUE->"
20 WAIT 49152, 128
30 PRINT"OK!"
]RUN
```

此例运行后，按任何一个键，程序就继续执行。因为当有键按下时，49152($C000)的最高位为1。

14. CALL

格式：CALL算术表达式

功能：调用以算术表达式的值为入口的机器语言程序。

方式：立即型与暂缓型。

例 1 CALL－936

此例是清除屏幕。

15. HIMEM:

格式：HIMEM:算术表达式

功能：设置BASIC程序运行时内存可使用的最高地址。

当刚进入BASIC时，系统自动设置了可使用的内存最高地址，其结果如下。

$ C000　没有DOS装入；

$ 9600　装入DOS后；

用户随时可用下述语句观察其结果

```
PRINT PEEK (116) *256+PEEK (115)
```

如果用户使用高分辨率绘图语句，通过执行

```
HIMEM:8191 (即$2000)
```

就可以保护图形不被程序破坏，若程序超过8191，计算机会告警：

```
? Out of memory error
```

16. LOMEM:

格式：LOMEM:算术表达式

功能：设置BASIC程序运行时，内存可用的最低地址。

方式：立即型与暂缓型。

如果用户不设置LOMEM:，则系统将其定为程序最末尾地址加4。当前LOMEM的值存放于106($6A)与105

($69) 单元中，可由PRINT PEEK(106)*256+PEEK(105)看到其结果。

17. PLAY

格式：PLAY

功能：将录到磁带上的程序装入内存并运行。

方式：立即型。

本语句是中华学习机的扩展BASIC语句，使用方法与LOAD类似，即先键入PLAY命令，同时按下录音机PLAY键（放音键）。当程序正确装入内存后会自动执行。

18. SASM

格式：SASM

功能：进入小汇编语言。

方式：立即型与暂缓型。

本语句是CEC-IA扩展的BASIC语句。当执行SASM之后，计算机就从BASIC语言方式下跳入小汇编语言（见小汇编的使用方法一章）方式下。提示符为“!”。用Ctrl-Reset可返回BASIC语言方式。

4.2.5 编辑与显示格式语句

1. LIST

格式：① LIST[行号1][-行号2]

　　　② LIST[行号1][，行号2]

功能：列出内存中的BASIC程序。

方式：立即型与暂缓型。

格式①与格式②的功能完全一样，在LIST语句中，如果行号1省略，则从第一条语句行开始列出；如果行号2省略，

则一直列到程序的最后一条语句行。

例 1 LIST（列出全部程序）。

例 2 LIST10—20（列出第10至第20行）。

例 3 LIST—300（从程序的头一行列到第300行）。

例 4 LIST25—（从第25行列至程序的最后一行）。

2. DEL

格式：DEL 行号1，行号2

功能：删除行号1至行号2之间的全部BASIC语句。

方式：立即型与暂缓型。

例 1 DEL 30，80

（删除第30行至第80行语句）

3. REM

格式：REM[字符串]

功能：注释BASIC程序。

方式：立即型与暂缓型。

REM只起注释作用，REM后面可跟任意字符，而不会被BASIC解释程序当作语句或命令执行。

例 1

```
10 REM PRINT"HAPPY NEW YEAR"
20 PRINT"HOW DO YOU DO!"
]RUN
HOW DO YOU DO!
```

4. HOME

格式：HOME

功能：清除屏幕，移光标到屏幕左上角。

方式：立即型与暂缓型。

HOME语句仅清除屏幕上的字符，而不破坏在内存的

BASIC程序。

5. VTAB

格式：VTAB 算术表达式

功能：把光标移到算术表达式的值指定的行上。

方式：立即型与暂缓型。

在西文方式下，表达式的值要在1～24范围内；在汉字状态下该值应在1～10范围内。

例 1

```
10  HOME
20  FOR I=1 TO 21 STEP 2: VTAB I
30  PRINT I : NEXT
40  END
```

此例在第1、3、5、…、21行上显示数字1、3、5、…、21。

6. HTAB

格式：HTAB 算术表达式

功能：把屏幕光标移到算术表达式的值给定的列上。

方式：立即型与暂缓型。

在西文40列方式下，每40列为一行，表达式可在0～255之间取值。在80列方式下，每80列为一行。在汉字34列方式下每34列为一行，但表达式仅可在1～34之间取值，超过此范围将出现位置错误，所以，汉字方式下 HTAB 仅对光标所在的那一行的列定位有效。

例 1：打印对角线（40列西文方式下）

```
10  HOME（清屏幕）
20  FOR I=1 TO 20
30  VTAB I（指定行的位置）
```

```
40  HTAB I+I-1 (指定列位置)
50  PRINT"*";
60  NEXT
]   RUN
*
 *
  *
   *
    *
     *
      *
       *
        *
         *
          *
           *
            *
             *
              *
               *
                *
                 *
                  *
                   *
```

7. CLEAR

格式：CLEAR

功能：清除任何变量及字符串等，并将各指针、堆栈归零，但不清除程序。

方式：立即型与暂缓型。

例1

```
10  X=128:Y=127
20  PRINT X, Y
30  CLEAR
40  PRINT X, Y
]RUN
128                 127
0                   0
```

8. FLASH

格式：FLASH

功能：使在屏幕上输出的字符闪烁。

方式：立即型与暂缓型。

注意FLASH语句仅对40列文本方式有效。

9. INVERSE

格式：INVERSE

功能：使在屏幕上输出的字符反相（即白底黑字）。

方式：立即型与暂缓型。

10. NORMAL

格式：NORMAL

功能：使屏幕上输出的字符正常显示（即黑底白字）。

方式：立即型与暂缓型。

在刚进入BASIC时，系统自动将字符输出方式设置为正常显示。在使用FLASH或INVERSE之后，可用NORMAL恢复正常显示方式。

例 1

```
10  HOME
20  INVERSE : PRINT"==";
30  FLASH : PRINT">";
40  NORMAL : PRINT"LOOK! "
```

4.2.6 定义语句

1. DIM

格式：DIM 变量名(下标)[{，变量名(下标)}]

功能：定义一个数组，并给数组在内存中开辟一个数据区。

方式：立即型与暂缓型。

数组中的变量称为下标变量，下标变量名的规定与普通变量名的规定相同。下标可以是正实数、零或正整数。下标的形式可以是数值、变量或算术表达式。下标必须放在圆括号内，且不能小于零。如果下标带有小数，将被自动取整。下面这些格式的下标都是可以的：

A(7)，A(2，3)，X(I，J)，Y(2*L，5+7)

DIM语句对数组的维数和每一维下标的上限和下限作出规定。下标的上限就是括号内的数值；下限全部设为零。维数由分隔下标上限的逗号确定。例如

A(15)

是一维数组，下标的下限为零，上限为15。又例如

A(100，50)

是二维数组，第一维下标的上限为100，第二维下标上限为50。

在使用数组变量之前，必须对数组定义，且数组变量的下标值不得超过上、下限规定的值。从语法规定上讲，数组允许的最大维数为88，但由于内存的限制，往往超出10维的数组已无法使用，如果使用DIM时出现

? Out of memory error

的错误时，表示内存已不够用，必须通过减少下标的上限或

减少维数校正之。此外，每一维的下标值不得超过32767。如果下标的上限不超过10，则可以不定义数组，直接使用数组变量。一个数组在程序中只能被定义一次，否则将出现

? Redim'd array error

的错误。

同简单变量一样，数组也有整型、实型和字符串型三种，例如：

DIM X%(15)，Y $(7)，C(99)

例 1 设有矩阵A和B，求A＋B？其中A和B分别为

$$A=\begin{pmatrix}1 & 3\\ 5 & 7\end{pmatrix},\quad B=\begin{pmatrix}2 & 4\\ 6 & 8\end{pmatrix}$$

程序：

```
10 REM ARRAY A+B->C
20 DIM A(1, 1), B(1, 1), C(1, 1)
30 FOR I=0 TO 1
40 FOR J=0 TO 1
50 READ A(I, J), B(I, J)
60 NEXT:NEXT
70 FOR I=0 TO 1
80 FOR J=0 TO 1
90 C(I, J)=A(I, J)+B(I, J)
100 PRINT C(I, J)
110 NEXT
120 PRINT
130 NEXT
140 DATA 1, 2, 3, 4, 5, 6, 7, 8
]RUN
3          7
11         15
```

2. DEF FN

格式：DEF FN 变量名(实型算术变量)＝算术表达式

功能：用户定义一个函数，使BASIC程序中直接调用这个函数。

方式：暂缓型。

使用DEF FN定义函数时要注意只能定义算术函数，不能定义字符串函数；使用自定义的函数之前必须定义函数；自定义函数是一元函数，自变量只能有一个；自定义函数不能递归定义。

例 1 计算：

$$1\times2\times3+2\times3\times4+\cdots+10\times(10+1)\times(10+2)$$

程序：

```
10  DEF FN MA(X)=X*(X+1)*(X+2)
20  FOR I=1 TO 10
30  S=S+FN MA(I)
40  NEXT
50  PRINT S
]RUN
4290
```

4.2.7 文本和音乐语句

1. TEXT

格式：TEXT

功能：使显示方式转换到文本显示方式。

方式：立即型与暂缓型。

CEC-ⅠA共有三种基本显示方式：

· 文本显示方式

·低分辨率图形方式

·高分辨率图形方式

每一种方式下又可以有单倍显示和双倍显示两种选择，因此CEC-ⅠA共有六种显示方式。

在文本方式下，单倍显示即通常所说的40列文本方式；双倍显示即通常所说的80列文本方式。

西文文本显示区在内存地址为：

$400～ $7FF

2. MUSIC

格式：MUSIC 算术表达式1，算术表达式2

功能：定义一个音阶和音长，并使机内的扬声器发音。

方式：立即型与暂缓型。

算术表达式1是发声的频率，取值范围（1～255），取值越大频率越低。算术表达式2是发声的时间，取值范围（1～255），取值越大发声时间越长。事实上，算术表达式1和算术表达式2还可以为0，但其代表取值256，因此MUSIC 0，0是频率最低、发声时间最长（约为1.3秒）的一个音。

MUSIC语句可以用于演奏音乐。其半音规律可用下述程序得到：

```
10  FOR I = 0 TO 24 STEP 0.5
20  T% = 256*(228/256)∧I:T% = T%*(T%<256)
30  PRINT T%; ", ";
40  NEXT
]  RUN
```

0，241，228，215，203，191，180，170，161，152，143，135，127，120，113，107，101，95，90，85，80，75，71，67，63，60，56，53，50，47，45，42，40，37，35，33，31，30，28，26，25，23，22，21，20，18，17，16，15

根据上表，用户可以确定多音程的音阶规律表。例如：取0(即256)为$\underset{\cdot}{5}$这个音，那么，其他音如表4-2所示。

表 4-2

取值	音阶	取值	音阶	取值	音阶	取值	音阶
0	$\underset{\cdot}{5}$	113	6	50	$\dot{7}$	23	$\dddot{1}$
228	$\underset{\cdot}{6}$	101	7	47	$\ddot{1}$	21	$\dddot{2}$
203	$\underset{\cdot}{7}$	95	$\dot{1}$	42	$\ddot{2}$	18	$\dddot{3}$
191	1	85	$\dot{2}$	37	$\ddot{3}$	17	$\dddot{4}$
170	2	75	$\dot{3}$	35	$\ddot{4}$	15	$\dddot{5}$
152	3	71	$\dot{4}$	31	$\ddot{5}$		
143	4	63	$\dot{5}$	28	$\ddot{6}$		
127	5	56	$\dot{6}$	25	$\ddot{7}$		

关于发声的时间即音阶的拍节，可根据音乐的速度来定。例如某一速度下取30为$\frac{1}{4}$拍，那么其他参数的节拍如下表所示。

声长取值	30	60	120	180	240
节　　拍	$\frac{1}{4}$拍	$\frac{1}{2}$拍	1拍	$1\frac{1}{2}$拍	2拍

例 1　以1拍演奏1、2、3、4、5、6、7、$\dot{1}$的旋律。

```
10  FOR 1 TO 8
20  READ T
30  MUSIC T, 120
40  NEXT
```

```
50  DATA 191, 170, 152, 143, 127, 113, 101, 95
]   RUN
```

4.2.8 扩充BASIC语句（&）

1、&

格式：&

功能：跳入 $3F5为入口的机器语言程序执行。

方式：立即型与暂缓型。

&语句与CALL语句都是转入机器语言程序，但由于CALL之后必须是入口地址，所以不便于扩充BASIC语句。而&语句之后允许接一些参数，只要在机器语言程序中加以一定处理即可。因此&语句在进一步扩充BASIC语句功能方面作用极大。使用&时应注意在 $3F5、$3F6、$3F7这三个单元放一条跳转指令（机器语言）

JMP 处理程序入口

而不能放处理程序本身，以免破坏后面内存单元的内容。

通常一个不超过120个字节的短的机器语言程序可以放在 $300（768）为首地址的内存单元中。长一些的机器语言程序可根据情况放到 $6000之后。

由于机器语言程序已超出了本书介绍的范围，下面仅给出一个使用&语句的例子，以说明&的使用方法，但不对机器语言的内容作说明。

首先进入监控状态，并输入下列程序：

```
03F5-  4C 00  03        JMP  $0300
0300-  20 67  DD        JSR  $DD67
0303-  20 52  E7        JSR  $E752
0306-  84 3C            STY  $3C
```

```
0308-   85 3D            STA $3D
030A-   A2 01            LDX #$01
030C-   4C 5E  FE        JMP $FE5E
```

此例是用 & 扩充 BASIC 语句，使用“& 算术表达式”的格式，反汇编20行机器语言。试执行

```
] &-151
```

执行此条语句后，可以看到从 $FF69到 $FF90一段的机器语言程序。

4.3 CEC-BASIC的函数

CEC-BASIC中有许多函数为用户提供。使用函数可以非常方便地解决程序中所遇到的各类问题。在使用函数时应特别注意，函数必须放在表达式内，而不能象语句那样单独使用，也不能给函数赋值。例如正弦函数SIN，象下面这样的使用都是可以的：

```
] PRINT SIN (0.5)
] X=SIN (0.5) +25
] IF SIN (0.1) <0.1 THEN END
```

但象下面这样的使用都是错误的：

```
] SIN (0.2)
] SIN (0.2) =0.27
```

下面将按照函数的类型对函数的作用作介绍。

4.3.1 三角函数

1. SIN（算术表达式）

正弦函数，其函数值等于算术表达式的值（弧度）的正弦。

2. COS（算术表达式）

余弦函数，其函数值等于算术表达式的值（弧度）的余弦。

3. TAN（算术表达式）

正切函数，其函数值等于算术表达式的值（弧度）的正切。

4. ATN（算术表达式）

反正切函数，其函数值等于算术表达式的值（弧度）的反正切。

上述这些三角函数的自变量均以弧度为单位。若要用度制时，需用下面这个公式转换成弧度后才可使用。

弧度＝角度数＊3.1415926/180

例如　求sin30°。

```
]  PRINT SIN(30*3.1415926/180)
0.500014426
```

4.3.2 算术函数

1. INT（算术表达式）

取整函数，其函数值等于或小于算术表达式的最大整数。

例

```
]  PRINT (21.87)
21
]  PRINT (0.45)

]  PRINT (-0.2)
-1
```

2. RND（算术表达式）

随机函数，若算术表达式的值大于0，则函数值是0～0.999999999之间的一个随机小数。

若算术表达式的值等于0，则函数值与上一次使用RND函数的值相等。

若算术表达式的值小于0，则函数值与算术表达式一一对应，也即相同的算术表达式，得到相同的函数结果。

例

```
10  FOR I=1 TO -1 STEP-1
20  FOR J=1 TO 3
30  PRINT"RND (";I;") =";RND (I)
40  NEXT
]  RUN
RND(1)=.738207502
RND(1)=.152381361
RND(1)=.376812451
RND(0)=.376812451
RND(0)=.376812451
RND(0)=.376812451
RND(-1)=2.99196472E-08
RND(-1)=2.99196472E-08
RND(-1)=2.99196472E-08
```

3. SGN（算术表达式）

符号函数，即$SGN(X)=\begin{cases}-1 & X<0\\ 0 & X=0\\ 1 & X>0\end{cases}$

例

```
10  READ X, Y, Z
20  PRINT SGN(X), SGN(Y), SGN(Z)
```

```
30  DATA 0.135, 0, -125
]   RUN
1             0             -1
```

4. ABS（算术表达式）

绝对值函数，即代数中的$|X|$

例：

```
]   PRINT  ABS(-2*7)
14
```

5. SQR（算术表达式）

平方根函数，即代数中的$\sqrt{X}$

例：

```
PRINT  SQR(2)
1.41421356
```

6. EXP（算术表达式）

指数函数，即代数中的e^x（其中e＝2.71828…）

例：

```
]   PRINT  EXP (1)
2.71828183
```

7. LOG（算术表达式）

对数函数，即代数中的$\log_e X$或$\ln X$

例

```
] PRINT LOG(2)
.693147181
```

若要计算常用对数$\log_{10}X$，可用换底公式：

$$\ln(x)/\ln(10)$$

例 计算$\log_{10}(100)$

```
] PRINT LOG(100)/LOG(10)
2
```

4.3.3 派生函数

通过前面介绍的函数，可以派生出更多的函数，利用自定义函数DEF FN即可定义它们，例如正割函数 SEC(X) 可按如下方法定义

例：

```
10  DEF FN SEC(X) = 1/COS(X)
20  PRINT FN SEC(0.5)
```

下面给出派生函数的定义：

正割：SEC(X) = 1/COS(X)

余割：CSC(X) = 1/SIN(X)

余切：COT(X) = 1/TAN(X)

反正弦：ARCSIN(X) = ATN(X/SQR(-X * X + 1))

反余弦：ARCCOS(X) = -ATN(X/SQR(-X * X + 1)) + 1.5708

反余切：ARCCOT(X) = -ATN(X) + 1.5708

反正割：ARCSEC(X) = ATN(SQR(X * X - 1)) + (SGN(X) - 1) * 1.5708

反余割：ARCCSC(X) = ATN(1/SQR(X * X - 1)) + (SGN(X) - 1) * 1.5708

双曲正弦：SINE(X) = (EXP(X) - EXP(-X))/2

双曲余弦：COSH(X) = (EXP(X) + EXP(-X))/2

双曲正切：TANH(X) = -EXP(X)/(EXP(X) + EXP(-X)) * 2 + 1

双曲正割：SECH(X) = 2/(EXP(X) + EXP(-X))

双曲余割：CSCH(X) = 2/(EXP(X) - EXP(-X))

双曲余切：COTH(X)=EXP(-X)/(EXP(X)
-EXP(-X))*2+1

反双曲正弦：ARGSINH(X)
=LOG(X+SQR(X*X+1))

反双曲余弦：ARGCOSH(X)
=LOG(X+SQR(X*X-1))

反双曲正切：ARGTANH(X)
=LOG((1+X)/(1-X))/2

反双曲正割：ARGSECH(X)
=LOG(SQR(-X*X+1)+1)/X

反双曲余割：ARGCSCH(X)
=LOG(SGN(X)*SQR(X*X+1)+1)/X

反双曲余切：ARGCOTH(X)
=LOG((X+1)/(X-1)/2)

A MOD B:MOD(A)=INT((A/B-INT(A/B))
*B+0.05)*SGN(A/B)

4.3.4 字符串处理函数

1. LEN（字符串表达式）

计算字串的长度，其值在0～255之间。

例

```
10  A$="CHINA"
20  B$="BEIJING"
30  PRINT LEN(A$), LEN(B$), LEN(A$+B$)
]   RUN
5                  7                  12
```

2. STR $（算术表达式）

将算术表达式的值转换成字符串。

例

```
10  A$=STR$(100+23)
20  B$="ABC"+A$
36  PRINT B$
]   RUN
ABC123
```

3. CHR $（算术表达式）

将算术表达式转换成与它的值对应的ASCII字符，算术表达式的值在0～255之间。

例

```
10  FOR I=65 TO 90
20  PRINT I, CHR$(I)
30  NEXT I
]   RUN
65                  A
66                  B
67                  C
68                  D
69                  E
70                  F
71                  G
72                  H
73                  I
74                  J
75                  K
76                  L
77                  M
78                  N
```

79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z

4. VAL（字符串表达式）

将字符串转换成数值，经转换后的数值可以参加算术运算，如果字符串表达式的第一个字符不是数字，则函数值为0；如果数字中带有字母字符，则字母后面的数字不参与转换。

例

```
10  X = VAL("123") + 1000
20  Y = VAL("A123")
30  Z = VAL("105A31")
40  PRINT X, Y, Z
]   RUN
1123                0                105
```

5. ASC（字符串表达式）

函数值为字符串表达式第一个字符的ASCII码。

例

```
10  X = ASC("A")
20  Y = ASC("BCD")
```

```
30  PRINT X, Y
]   RUN
65                  66
```

6. LEFT $（字符串表达式，算术表达式）

得到字符串最左段的m个字符，m是算术表达式的 值。

例

```
10  A$="ABCDEFG"
20  B$=LEFT$(A$, 4)
30  C$=LEFT$(A$, 2)
40  PRINT A$, B$, C$
]   RUN
ABCDEFG             ABCD            AB
```

7. RIGHT $（字符串表达式，算术表达式）

得到字符串最右段的m个字符，m是算术表达式的值。

例

```
10  A$="ABCDEFG"
20  B$=RIGHT$(A$, 3)
30  PRINT B$
]   RUN
EFG
```

8. MID $（字符串表达式，算术表达式1，[算术表达式2]）

将字符串表达式的第n个字符至第n+m－1个字符取作函数值。其中n是算术表达式1，m为算术表达式2。若算术表达式2省略，则将字符串表达式的第n个字符串向右取，直至字符串的末尾。

例

```
10  A$="ABCDEF"
20  B$=MID$(A$, 2, 3)
```

```
30  C$=MID$(A$, 5)
40  PRINT B$, C$
]   RUN
BCD                  EF
```

4.3.5 数制转换函数

1. DEC（字符串表达式）

此函数是CEC-ⅠA扩充的标准函数，其功能是将字符串表达式给出的十六进制字符串，转换成十进制算术值。其中字符串的范围是（$0000～$FFFF），函数值的范围是（-32768～32767）。

例

```
PRINT DEC ("FF69")
-151
 ALL DEC ("FC58")
```

清屏幕

```
CALL DEC ("FE80")
```

置反相显示方式。

2. HEX $（算术表达式）

此函数是CEC-ⅠA扩充的标准BASIC函数，其功能是把算术表达式的值转换成十六进值形式的字符串。它与DEC函数互为反函数。算术表达式的取值范围是：

-65535～+65535

超过此范围的数按65536取模。函数值的范围是$0000～$FFFF。

例 1

```
PRINT HEX$(255)
FF
```

例 2

```
10  A$=HEX$(195):B$=HEX$(16)
20  ? A$, B$, A$+B$
30  ? DEC(A$+B$)
RUN
C3                 10                  C310
-15600
```

4.3.6 其它函数

1. TAB（算术表达式）

TAB函数是控制打印格式的函数。使用这个函数可以使打印机的打印头向右移到算术表达式所给的值的那一列上。例如TAB(20)表示把当前打印头移到第20列，但如果当前打印头已超过第20列，则打印头仍在当前位置不动。

算术表达式的取值范围是0～255，其中0代表取值256。

如果打印机没有接通（通常用PR#1接通打印机），则字符只显示在屏幕上，其效果相当于光标对应打印头，只不过当算术表达式超过40时，就转到下一行去显示字符。

2. POS（算术表达式）

这个函数能取到当前光标所在列的位置。其中算术表达式仅为了满足标准函数格式虚设的数值，与函数的结果无关，通常取0即可。

函数值的范围是0～39，0表示当前光标在最左端，39表示当前光标在最右端。

3. SPC（算术表达式）

这个函数只能用在PRINT语句中，其功能是以上一次输出的最后一个字符位置为基准，跳过若干个空格。跳过的

空格数由算术表达式的值确定。算术表达式的取值范围是0～255。

例

```
]  PRINT "A" SPC(2) "B" SPC(3) "C"
    A   B    C
]  PRINT SPC(255)SPC(200)
```

空出一大片空格。

4. FRE（算术表达式）

这个函数的功能是释放数据区中无用的单元，使其能为新的数据使用。

算术表达式的值不影响函数功能，可任意选择，如取0或1。函数返回的值是变量可以使用的内存数目，但其结果通常是带符号数，不习惯此种表示方法的用户可用下面的方法转换：

```
PRINT  65536+FRE(0)
```

习惯于十六进制表示的用户可由

```
PRINT  HEX $(FRE(0))
```

得到。

定期使用FRE函数可以使内存的使用效率提高。

5. PDL（算术表达式）

此函数能得到游戏控制器第n个摇杆的当前值。n是算术表达式的值，其范围是0～3，如果超过3则以4取模。函数值在0～255之间，每一个值与控制器中的可变电阻值有一一对应的关系。

如果有连续两个PDL读入二个游戏控制器的值，则需在两个PDL之间加入一个短的延时程序（如FOR I=1 TO 10:NEXT），以免读取的结果相互影响，例如

```
10  X=PDL(0)
20  FOR I=1 TO 10:NEXT
30  Y=PDL(1)
⋮
```

除了用PDL可读出4个不同的游戏控制器的值外，还可用PEEK函数读出3个游戏按钮（即开关量的输入），其地址为 $C061～ $C063,当读出的数据的最高位为1(即≥128)时，表示有开关按下；为0表示没有开关按下。

例如：当第1个按钮按下时，输出“CEC-ⅠA”

```
10 IF PEEK(DEC("C061"))<128 THEN 10
20 PRINT"CEC-IA"
```

6. SCRN（算术表达式1，算术表达式2）

此函数能得到屏幕上一点的低分辨率图形的颜色代码。屏幕上点的坐标由算术表达式1(X坐标）和算术表达式2（Y坐标）给出。其中0≤X≤39，0≤Y≤47。

7. PEEK（算术表达式）

这个函数能读取以算术表达式的值为地址的内存单元的内容。

例如：

```
] PRINT PEEK (DEC("C061"))
```

显示第一个游戏按钮的当前值。

4.3.7 扩充函数的函数

1. USR（算术表达式）

与扩充BASIC语句的语句&类似，CEC-BASIC也为用户提供了扩充函数的函数，即USR。

这个函数能将算术表达式的值放到BASIC使用的浮点

累加器里，并转到 $000A执行机器语言。通常在$0A～$0C中存放一个JMP指令，以转入用户编写的扩充函数的处理程序。

用户在编写扩充函数时应注意，函数的自变量已由BASIC解释程序放到浮点累加器（$9D～$A3），函数的值应由用户的机器语言程序放回到浮点累加器中，并用RTS指令返回BASIC解释程序。

由于机器语言的编程已超出了本书的介绍范围，这里只提供三个小例子，以说明USR的使用方法，有兴趣的读者可参考有关6502机器指令的手册，深入了解机器语言的编程方法。

例 1

机器语言部分：

```
000A-60   RTS
```

BASIC语言部分：

```
]   PRINT USR(256)
256
```

说明：此程序很简单，我们作的函数就相当于f(x)=x，也就是说自变量与函数值相等。在 $0A～ $0C中仅存放了一条返回指令，这说明浮点累加器没有修改，所以函数的值还是浮点累加器原来的值，即自变量本身。

例 2

机器语言：

```
000A-4C  00  03    JMP $0300
0300-20  63  EB    JSR  $EB63
0303-4C  87  E9    JMP $E987
```

BASIC语言：

```
]  PRINT USR(3)
 9
10  FOR I=1 TO 10
20  PRINT USR(I), ", ";
30  NEXT
]  RUN
1, 4, 9, 16, 25, 36, 49, 64, 81, 100,
```

此例扩充的USR函数是平方函数，即$f(x)=x^2$。其中机器语言子程序 $EB63是把浮点累加器搬到另一个浮点累加器的暂存单元，子程序 $E987是将上述两个浮点累加器相乘，并把相乘的结果放回到浮点累加器里返回。

例 3

机器语言：

```
000A-   4C 00 03      JMP $0300
0300-   A9 13         LDA #$13
0302-   A0 E9         LDY #$E9
0304-   4C 66 EA      JMP $EA66
```

BASIC语言

```
]  PRINT USR(2), USR(7)
.5                   .142857143
]  PRINT USR(3), USR(4)
.333333333           .25
```

此例利用USR函数扩充了倒数函数，也即$f(x)=1/x$。

通过上述三例可以看出，USR函数的作用是非常大的。特别是当用户想使用某种特别函数，而现在的 BASIC函数里又没有提供，则USR函数为用户扩充函数提供了方便。

4.3.8 范例——赛马游戏程序

为使用户进一步掌握发音语句、显示格式语句、基本转

移语句以及字符串处理函数、随机函数和数组等的使用方法，我们为用户提供了一个综合性的范例——赛马游戏程序。

程序中用反相的数字代表奔跑的马，一共有六匹马。它们同时起跑，并请你猜一猜哪匹马最先跑到终点（屏幕的最右端）。在马奔跑时，跑道两旁的栅栏在飞速向后移动，以显示马跑得很快。在屏幕的右下角还有一个秒表在计时（秒表的时间未经校对）。当某一匹马率先到达终点时，计算机告诉你获胜的是哪一匹马，请看看你猜中了吗？

```
10  DIM A(6), W$(4)
20  HOME:PRINT"HORSE:"
30  INVERSE:FOR I=1 TO 6:VTAB I*3:PRINT I:NEXT
40  NORMAL:PRINT SPC(45)"GUESS WHICH HORSE CAN
    WIN? "; :GET A$
50  FOR I=1 TO 11:W$=W$+"1"+"==":NEXT
60  FOR I=1 TO 4:W$(I)=MID$(W$, I, 40):NEXT
70  FOR I=1 TO 6:MUSIC I*40, 39:NEXT
80  NORMAL:VTAB 22:HTAB 28:PRINT"TIME:"
90  VTAB 22:HTAB 33:PRINT T:T =T+1
100  FOR I=1 TO 4
110  VTAB 1:HTAB 1:PRINT W$(I)
120  VTAB 20:PRINT W$(I)
130  N=INT(6*RND(1))+1:A(N)=A(N)+1:M=A(N)
140  VTAB N*3:HTAB M:PRINT " ";
150  INVERSE:HTAB M+1:PRINT N
160  NORMAL:MUSIC 110, 1
170  IF M>38 THEN 190
180  NEXT:GOTO 90
190  VTAB 21:PRINT:PRINT"NUMBER"; N; "WINS! "
200  MUSIC 200, 22:END
```

4.4 图形语句

4.4.1 低分辨率绘图语句

1. GR

格式：GR

功能：把屏幕设置为低分辨率图形方式（40×40）并在屏幕下方留4行作为文本窗口。

方式：立即型与暂缓型。

执行了GR语句之后，屏幕窗口转为图形方式，即可通过其它低分辨率绘图语句，执行绘图命令。GR语句使COLOR设置为0，同时清除屏幕为黑色。

如果不想把屏幕最下面的4行作为文本行，可在GR语句之后执行

POKE -16302，0

或 POKE 49234，0。

执行

POKE -16301，0

或 POKE 49235，0

可以再返回混合显示方式。

2. COLOR

格式：COLOR=算术表达式

功能：设置低分辨率绘图的颜色。

方式：立即型与暂缓型。

COLOR语句中的算术表达式可在0～255之间取值，但有效值范围是0～15，超过此范围的数值按16取模。例如

COLOR = 17

与

COLOR = 1

等效。算术表达式的值所表示的颜色如下

0	黑色	1	红色	2	深蓝色	3	紫红色
4	深绿色	5	深灰色	6	中蓝色	7	浅蓝色
8	棕色	9	橙色	10	灰色	11	粉红色
12	绿色	13	黄色	14	绿蓝色	15	白色

3. PLOT

格式：PLOT 算术表达式1，算术表达式2

功能：画一个点。

方式：立即型与暂缓型。

PLOT语句以算术表达式1为X坐标，算术表达式2为Y坐标，以刚由COLOR语句指定的颜色画一点。

例

```
10  GR
20  COLOR = 15:REM  WHITE  COLOR
30  PLOT 20, 20
]   RUN
```

坐标的取值范围是

$0 \leqslant X \leqslant 39$，$0 \leqslant Y \leqslant 47$

4. HLIN

格式：HLIN 算术表达式1，算术表达式2
        AT 算术表达式3

功能：画一条水平线。

方式：立即型与暂缓型。

HLIN语句以算术表达式1和算术表达式3为起点的X、Y坐标，以算术表达式2和算术表达式3为终点的X、Y坐标

画一条直线。

例

```
10  GR
20  COLOR=1
30  HLIN 15, 30 AT 20
]   RUN
```

此例从（15，20）到（30，20）画一条红色直线。

5。VLIN

格式：VLIN 算术表达式1，算术表达式2
　　　　AT 算术表达式3

功能：画一条垂直线。

方式：立即型与暂缓型。

VLIN语句以算术表达式3和算术表达式1为起点，以算术表达式3和算术表达式2为终点画一条垂直线。

例 1

```
10  GR
20  COLOR=2
30  VLIN 10, 30 AT 20
]   RUN
```

此例以（20，10）为起点，以（20，30）为终点画一条竖线。

例 2

```
10  GR:COLOR=15
20  HLIN 10, 30 AT 10
30  HLIN 10, 30 AT 30
40  VLIN 10, 30 AT 10
50  VLIN 10, 30 AT 30
]   RUN
```

此例在屏幕上画一矩形框。

例 3

```
10  GR:COLOR=15
20  A=30/48
30  R=15
40  PI=3.1415926
50  FOR S=0 TO PI+PI STEP PI/36
60  X=A*R*COS(S)
70  Y=R*SIN(S)
80  PLOT 20+X, 20+Y
90  NEXT
100  END
]  RUN
```

例3是以（20，20）为圆心，以15为半径画一个圆环。改变第20行语句中的变量A的值，可以画出各种椭圆。

4.4.2 高分辨率绘图语句

CEC-ⅠA型中华学习机除具备低分辨率绘图功能外，同时还具备高分辨率（280×192点）绘图功能。此外，在倍高分辨率（560×192点）方式下也可绘出丰富多采的画面，但CEC-BASIC尚未提供在倍高分辨率下的绘图语句，有兴趣的用户可自己开发倍高分辨率绘图软件。

CEC-ⅠA计算机为高分辨率绘图开辟了两个页面，第一页占用内存 \$2000～\$3FFF一段空间，第二页占用内存 \$4000～\$5FFF一段空间。用户可通过

POKE -16300, 0

或 POKE 49236, 0

选择第一页，通过

POKE －16299，0

或 POKE 49237，0

选择第二页。

CEC-ⅠA型计算机还可以使屏幕处于混合显示方式，在混合方式下，屏幕最下面的4行留作文本显示。

用户可通过

POKE －16302，0

或 POKE 49234，0

进入全图形方式；通过

POKE －16301，0

或 POKE 49235，0

进入混合图形方式。

此外，所有绘图显示方式均以屏幕的左上角为坐标原点。

1. HGR

格式：HGR

功能：置高分辨率第一页混合图形方式，并清除原第一页内的图形。

方式：立即型与暂缓型。

执行HGR后，屏幕清为黑色，但不影响HCOLOR设定的颜色。

2. HGR2

格式：HGR2

功能：置高分辨率第二页全图形方式，并清除原第二页内的图形。

方式：立即型与暂缓型。

执行HGR2后，屏幕清为黑色，且不影响HCOLOR设定的颜色。

3. HCOLOR

格式：HCOLOR＝算术表达式

功能：设置高分辨率绘图方式下的颜色。

方式：立即型与暂缓型。

算术表达式的取值范围在0～7之间。每个值所对应的颜色如下：

0 黑1　1 绿　2 紫　3 白1　4 黑2　5橙　6蓝　7 白2

4. BCLR

格式：BCLR 算术表达式

功能：将整个高分辨率图形页面置成某一种颜色。

方式：立即型与暂缓型。

BCLR是CEC-I A BASIC扩充的一条语句。它可以把当前高分辨率图形方式下的整个屏幕全都填上某一种颜色，填充的颜色由算术表达式的值来确定。算术表达式的取值范围及每一个值所代表的颜色都与HCOLOR语句完全一样。

例

```
HGR2 : BCLR6
```

此例把整个屏幕染成蓝色。

BCLR语句可用于绘图之前上底色用。

5. HPLOT

格式：①HPLOT 算术表达式1，算术表达式2

②HPLOT TO 算术表达式1，算术表达式2

③HPLOT 算术表达式1，算术表达式2

TO 算术表达式3，算术表达式4

[{TO 算术表达式，算术表达式}]

功能：在高分辨率绘图方式下画一个点或画一些直线。

方式：立即型与暂缓型。

在解释上述三格式的HPLOT语句之前，先作一些说明：格式中用逗号分开的每对算术表达式均表示了一个坐标，这里将用X和Y代表它们，其取值范围如下：

$0 \leqslant X \leqslant 279$，$0 \leqslant Y \leqslant 191$

第一种格式：

HPLOT X，Y

在（X，Y）处画一个点。

例 1

```
10  HGR:HCOLOR=1
20  HPLOT 100，100
]   RUN
```

第二种格式：

HPLOT TO X，Y

从当前的坐标点到（X，Y）点画一条直线。如果这个语句之前没有画过任何点或线，则这条命令不会画出线来。

例 2

```
10  HGR:HCOLOR=1
20  HPLOT 100，100
30  HPLOT TO 0，0
]   RUN
```

此例把坐标点（100，100）与（0，0）连成一条绿色直线。

第三种格式：

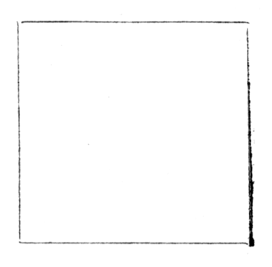
HPLOT X_1，Y_1 TO X_2，Y_2 TO X_3，Y_3…

将（X_1，Y_1）到（X_2，Y_2）到（X_3，Y_3）…用直线连

接起来。

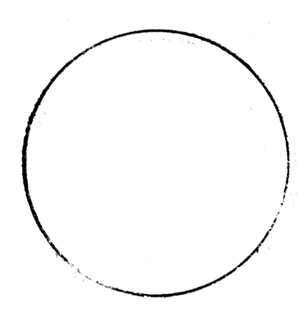
例 3

```
10  HGR:HCOLOR=7
20  HPLOT 0, 0 TO 99, 0 TO 99, 99 TO 0, 99 TO 0, 0
]   RUN
```



例 4

```
10  HGR:HCOLOR=7
20  R=55
30  HPLOT R+100, 100
40  FOR S=0 TO 6.484 STEP 0.2
50  HPLOT TO R*COS(S)+100, R*SIN(S)+100
60  NEXT
]   RUN
```



4.4.3 高分辨率向量绘图语句

CEC-BASIC有4个处理高分辨率向量绘图语句，它们

是

DRAW、XDRAW、ROT、SCALE

在使用这些语句之前，先要用图形表把图形定义出来。

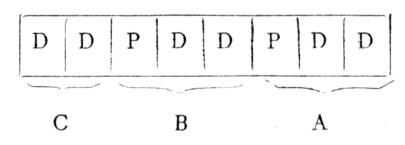
图形表由图形索引表和图形定义表两部分组成。索引表的结构如下：

内存地址	内容
S	图形定义的总个数（≤255）
S+1	未用
S+2 S+3	第一个图形定义的首地址与S的偏移量（相对地址）
S+4 S+5	第二个图形定义的首地址与S的偏移量
⋮	
S+2n S+2n+1	第n个图形定义的首地址与S的偏移量

此表中，S代表内存的某一个地址，如S＝ $300，表中右边一栏指明该地址所指向的内存中的内容。偏移量共占两个字节，头一个字节放偏移量的低位，次一个字节放偏移量的高位。

图形的定义表是由每一个图形的图形描述向量组成的。图形的描述方法是这样的：

首先把每个字节分为三段：



对D和P的规定如下

DD　含义

00　↑

01　→　　　　$P=\begin{cases}0 & \text{不画点}\\1 & \text{画点}\end{cases}$

10　↓

11　←

由于C段中没有P栏，因此机器内部规定P只能是0。

在填写图形的描述向量时，应注意：

① 图形向量存入内存时，先填 A 段，再填 B 段，最后填C 段。如果图形没有存完，可在下一个字节的 A 段开始，继续填写。

② 如果C段为00，表示C段不起作用，如果C段为00，B段为000，则表示C、B两段均不起作用。

③ 每一个图形描述的最后一个字节必须为0，以表示图形结束。

当使用图形表时，用监控的数据存储方法（详见监控的使用方法一章）或POKE语句把图形表存入内存（为避免破坏BASIC程序，通常首地址取 $6000），然后要把图形表的首地址放入232和233（即 $E8，$E9）两个单元，存放时要先放地址的低位（例如，图形表首地址为 $6000，则 *E8:00 60，或POKE 232，0　POKE 233，96）

下面将逐个介绍向量绘图语句，并通过具体的例子，进一步说明图形表的编制方法。

1. DRAW

格式：①DRAW 算术表达式1　AT算术表达式2，算

术表达式3

②DRAW 算术表达式1

功能：根据图形表绘图。

方式：立即型与暂缓型。

算术表达式1代表要画图形定义表中的第n个图形。

如果DRAW采用第一种格式，则表示以算术表达式2和算术表达式3为起点坐标，画出一个图形。

如果采用第二种格式，则在上次画完HPLOT、DRAW、XDRAW的位置处，画出一个定义图形。

例 画一个十字标，如图4-1(a)。

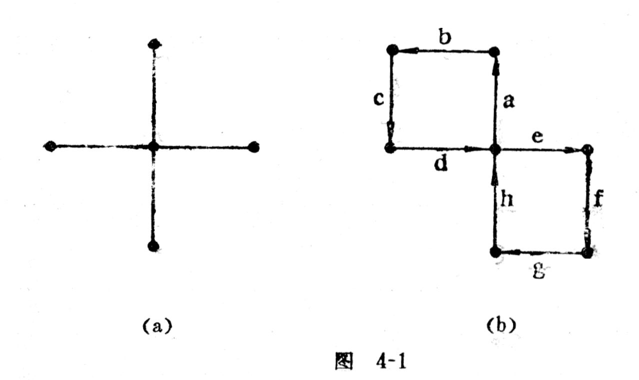


图 4-1

要画出图4-1(a)中的5个点组成的图形，可以用图4-1(b)中表示的a～h的八个步骤来完成。a～h对应的图形向量分别为：

a：100　　b：011　　c：010　　d：101

e：101　　f：010　　g：011　　h：100

图形描述为：

C		A	十六进制
10	011	100	9C
10	101	101	AD
00	100	011	23
00	000	000	00

将此描述向量写成图形定义的图形表就是：

S	$01	共有一个图形
S+1	$00	
S+2	$04	偏移量：4
S+3	$00	
S+4	$9C	图形定义
S+5	$AD	
S+6	$23	
S+7	$00	

利用BASIC语句把图形表存入内存 $6000并用DRAW画出这个十字标的程序如下。

```
10  HGR
20  POKE 232, 0        } 指定图形表首地址
30  POKE 233 , 96
40  FOR I=24576 TO 24583
50  READ X$
60  POKE I, DEC(X$)
70  NEXT
80  DATA 1, 0, 4, 0, 9C, AD, 23, 0
100  DRAW 1 AT 99, 99
]   RUN
```

2. SCALE

格式：SCALE=算术表达式

功能：确定DRAW和XDRAW所画图形的比例尺寸。

方式：立即型与暂缓型。

在DRAW的例1中，所画的十字标大小不能确定。用SCALE语句可以规定其大小。算术表达式在0～255之间取值。SCALE＝1 图形尺寸最小，数值越大比例就越大。如果取0，代表取值256，是最大的一个比例。

为节省篇幅，在以下的图形向量绘图语句的举例中，均以DRAW语句的例1所作的图形表为例，不一样的语句从第100行开始，第10至第80行语句的内容不变。

例 按放大4倍的尺寸画十字标。

```
100  SCALE=4
110  DRAW 1 AT 20, 20
]    RUN
```

3. XDRAW

格式：①XDRAW 算术表达式1 AT算术表达式2，算术表达式3

②XDRAW 算术表达式1

功能：用DRAW画的图形颜色的补色画一图形。

用XDRAW这个语句与用DRAW一样画一个图形，但用的是补色画图。下面每对颜色互为补色：

黑色-白色　蓝色-绿色　橙色-紫色

用互补的颜色在同一位置画同一个定义图形，就可以擦掉该位置上的那个图形。

例 1 让十字光闪烁。

```
100  SCALE=4
110  DRAW 1 AT 100, 100
120  FOR I=1 TO 300:NEXT
130  XDRAW 1 AT 100, 100
```

```
140   OR I=1 TO 300:NEXT
150 GOTO 110
```

例 2 让十字标向右移动。

```
100 SCALE=4
110 FOR X=5 TO 270
120 DRAW 1 AT X, 100
130 FOR I=1 TO 30:NEXT
140 XDRAW 1 AT X, 100
150 NEXT
]  RUN
```

4. ROT

格式：ROT=算术表达式

功能：设置待画图形的旋转角度。

方式：立即型与暂缓型。

旋转角度由算术表达式确定，其值在0～255之间，有效值以64取模。此外，旋转角度还与SCALE中的值有关。设

SCALE=n，ROT=m

则

n= 1：
- m=0　顺时针旋转0度
- m=16　顺时针旋转90度
- m=32　顺时针旋转180度
- m=48　顺时针旋转270度

n= 2：根据m可识别8种旋转度

n= ≥5：根据m可识别64种旋转度

例 1 画一个旋转的十字标。

```
100 SCALE=20
110 FOR X=1 TO 255
```

```
120  ROT = X
130  DRAW 1 AT 100, 100
140  XDRAW 1 AT 100, 100
150  NEXT
]  RUN
```

例 2 画一个右移并在旋转的十字标。

```
100  SCALE = 20
110  FOR X = 15 TO 255
120  ROT X
130  DRAW 1 AT X, 100
140  XDRAW 1 AT X, 100
150  NEXT
160  DRAW 1 AT X, 100
]  RUN
```

第五章 监控与小汇编的使用方法

5.1 监控的进入与退出

当CEC－ⅠA型计算机开机后，先进入选择单状态，等待用户选择BASIC或监控。选择单的第四项是：

3.MONITOR

用户按数字键“3”便可进入监控系统，若在BASIC语言系统下，使用

CALL-151

亦可进入监控系统。

进入监控后，系统的提示符为

*

若用户希望从监控系统中退到BASIC系统，可使用下列方法之一。

① 按Ctrl和Reset键

② 按Ctrl-C键，再按回车键

③ 按Ctrl-B键，再按回车键（清除了BASIC程序）

④ 按F6键，进入选择单去选择要进入的系统

5.2 监控命令

在进入监控系统以后，出现*符号的提示符，此后，监控系统便可以接受监控提供的命令。通常监控命令必须从键盘输入，当监控接受到监控的命令及一个回车（Return）键之后，立即执行此命令。

下面将逐条介绍监控的各种命令。

5.2.1 显示存储器的内容

格式①：〈地址〉

功能：显示〈地址〉单元的内容。

例

```
 *FF69
FF69 - A9
```

格式②：〈地址1〉·〈地址2〉

功能：显示从地址1到地址2之间所有单元中的内容。

例

```
*F800.F83F

F800-  4A  08  20  47  F8  28  A9  0F
F808-  90  02  69  E0  85  2E  B1  26
F810-  45  30  25  2E  51  26  91  26
F818-  60  20  00  F8  C4  2C  B0  11
F820-  C8  20  0E  F8  90  F6  69  01
F828-  48  20  00  F8  68  C5  2D  90
F830-  F5  60  A0  2F  D0  02  A0  27
F838-  84  2D  A0  27  A9  00  85  30
*
```

格式③：.〈地址2〉

功能：显示从当前地址至地址 2 之间内存单元的内容。

例

```
*.F880

F840- 20  28  F8  B8  10  F6  60  48
F848- 4A  29  03  09  04  85  27  68
F850- 29  18  90  02  69  7F  85  26
F858- 0A  0A  05  26  85  26  60  A5
F860- 30  18  69  03  29  0F  85  30
F868- 0A  0A  0A  0A  05  30  85  30
F870- 60  4A  08  20  47  F8  B1  26
F878- 28  90  04  4A  4A  4A  4A  29
F880- 0F
```

格式④：只按一个回车（RETURN）键

功能：显示从当前地址至地址最低位为 7 或F的内存单元中的内容。

当检查了某个单元的内容之后，如还想继续检查下面的一些存储单元的内容，可按回车键，则监控会继续显示下面一些单元的内容，显示的结束地址为×××7或×××F。

例

```
*FF69
FF69- A9
*（按回车键）
AA 85 33 20 67 FD
*
```

5.2.2 改变存储单元中的内容

格式①：〈地址〉:数据␣数据␣…

功能：从地址单元开始顺序将数据存入内存单元（符号“␣”表示空格键，下同）。

例

```
*8000:FF FF
*8000（回车）
FF
```

格式②：:数据␣数据␣…

功能：从当前存储地址开始顺序将数据存入内存。

5.2.3 移动存储器中的内容

格式：〈地址 1〉＜〈地址 2〉.〈地址 3〉M

功能：将地址 2 至地址 3 之间的全部内存单元中的内容复制到地址1为起始单元的内存中去。

5.2.4 核实存储器中的内容

格式：〈地址 1〉＜〈地址 2〉.〈地址 3〉V

功能：核实从地址 2 到地址 3 之间的数据是否与从地址开始长度相等的数据块相同。若有不同，就显示出来。

5.2.5 从磁带中读入数据块

格式：〈地址 1〉.〈地址 2〉R

功能：将磁带中的数据读入内存，并放到地址 1 到地址之间的存储单元中。磁带中数据长度必须与〈地址 2〉－〈地址 1〉＋1的值相等。

5.2.6 将数据块写入磁带

格式：〈地址 1〉.〈地址 2〉W

功能：将〈地址 1〉至〈地址 2〉之间的数据写到磁带

上。

5.2.7　置屏幕反相显示方式

格式：I

功能：置屏幕为反相（即白底黑字）显示方式。

5.2.8　置屏幕正常显示方式

格式：N

功能：置屏幕为正常（即黑底白字）显示方式。

5.2.9　显示及修改CPU寄存器

格式：Ctrl-E

功能：显示累加器A，变址寄存器X、Y，状态寄存器P及堆栈指针S的内容。如果要修改上述寄存器的内容，可在键入:之后，用空格为分隔符，键入修改的寄存器值。

5.2.10　反汇编命令

格式①：〈地址〉L

功能：从指定地址开始反汇编20条机器语言指令。

例

```
*F800L

F800-  4A               LSR
F801-  08               PHP
F802-  20   47   F8     JSR   $F847
F805-  28               PLP
F806-  A9   0F          LDA   #$0F
F808-  90   02          BCC   $F80C
F80A-  69   E0          ADC   #$E0
```

```
F80C-   85   2E        STA   $2E
F80E-   B1   26        LDA   ($26), Y
F810-   45   30        EOR   $30
F812-   25   2E        AND   $2E
F814-   51   26        EOR   ($26), Y
F816-   91   26        STA   ($26), Y
F818-   60             RTS
F819-   20   00   F8   JSR   $F800
F81C-   C4   2C        CPY   $2C
F81E-   B0   11        BCS   $F831
F820-   C8             INY
F821-   20   0E   F8   JSR   $F80E
F824-   90   F6        BCC   $F81C
*
```

格式②：L

功能：从当前地址开始反汇编20条机器语言指令。

例

```
*L
F826-   69   01        ADC   #$01
F828-   48             PHA
F829-   20   00   F8   JSR   $F800
F82C-   68             PLA
F82D-   C5   2D        CMP   $2D
F82F-   90   F5        BCC   $F826
F831-   60             RTS
F832-   A0   2F        LDY   #$2F
F834-   D0   02        BNE   $F838
F836-   A0   27        LDY   #$27
F838-   84   2D        STY   $2D
F83A-   A0   27        LDY   #$27
```

```
F83C-   A9    00        LDA   #$00
F83E-   85    30        STA   $30
F840-   20    28  F8    JSR   $F828
F843-   88              DEY
F844-   10    F6        BPL   $F83C
F846-   60              RTS
F847-   48              PHA
F848-   4A              LSR
```

5.2.11 执行机器指令

格式①：〈地址〉G

功能：从指定地址开始执行机器语言的程序。

例

*FC 58 G

（$FC 58是清屏幕并把光标移到左上角的机器语言子程序。）

格式②：Ctrl-Y

功能：从地址 $3F8开始执行机器语言的程序。通常$3F8处放一条JMP指令，转向实际要执行的程序。

例

*3F8：4C　60　FB　(JMP　$FB60)

*Ctrl-Y　回车

显示：

CHINA EDUCATION COMPUTER

VERSION 2.0

5.2.12 选择输入设备

格式：〈槽号〉Ctrl-K

功能：将输入控制转给指定槽口。

5.2.13 选择输出设备

格式：〈槽号〉Ctrl-P

功能：将输出控制转给指定的槽口。

槽口1、2、3、6已由系统使用，其作用如下：

1 号槽：打印机接口

2 号槽：通讯接口

3 号槽：根据 $2FE单元中的内容选择相应的显示方式。

$$
\$2FE=\begin{cases}0，40列文本方式\\1，汉字图形方式\\2，80列文本方式\end{cases}
$$

6 号槽：磁盘驱动器接口。

槽口4、5、7可提供给用户使用。

5.2.14 十六进制加减法运算

格式①：〈数据 1〉+〈数据 2〉

功能：执行单字节十六进制加法运算。

例

```
*A+10
=1A
```

格式②：〈数据 1〉-〈数据 2〉

功能：执行单字节十六进制减法运算。

例

```
*1A-10
=0A
```

在加减运算中，其结果也是单字节，相当于其结果按256取模。

5.2.15 多重命令

监控系统允许在一个命令行内写入多个监控命令，只要总的字符个数小于254即可。如果多重命令是单字母 命 令，则字母之间不需用分隔符，例如

```
*LLL
```

表示执行三次反汇编20行指令的命令，因此，它可以显示60行指令助记符。

如果多重命令是修改存储器内容的命令，则以N作为分隔符，例如

```
*1000:FF N 100F:AA BB
```

其它情况下，可用空格作为分隔符。

5.3 控制键Contrl的功能

在40列和80列文本方式下，Contrl键与下列字母（或字符）键之一同时按下，可产生一定的作用。

字母	功能	注释
G	产生1kHz的单音0.1秒	
H	光标左移一格	
J	光标下移一行	
K	从当前光标清至屏幕底端	1
L	清屏幕，光标至左上角	1
M	回车，即Return键	
N	置正常显示方式	1，3
O	置反相显示方式	1，3
Q	置40列显示	1
R	置80列显示	1

S	暂停显示，直至有一个键按下	1，2
U	退出80列控制系统	1
V	向下滚屏一行	1
W	向上滚屏一行	1
Y	将光标移至左上角	1
Z	清光标所在的那一行	1
\	光标右移一格	1
]	从当前光标清至行尾	1

注释解释：

1. 在80列系统下才有效
2. 只能从键盘输入
3. 不能从键盘输入，要通过程序输入。

5.4 屏幕编辑键Esc的功能

在40列文本及80列文本方式下，按Esc键之后，再按下列字母或字符之一，可对文本进行屏幕编辑。通过屏幕编辑，可以修改BASIC和其它一些程序。

字符	功能	注释
@	清屏幕，光标移至左上角	
A	光标上移一行	
B	光标右移一格	
C	光标左移一格	
D	光标下移一行	
E	从当前光标清至行尾	
F	从当前光标清至屏幕底端	
I或↑	光标上移一行	2
J或←	光标左移一格	2
K或→	光标右移一格	2

M或←	光标下移一行	2
R	接通强制上档方式	1
T	断开强制上档方式	1
4	进入40列显示方式	1
8	进入80列显示方式	1
Ctrl-Q	退出80列控制系统	1

注解：

1.在80列控制系统下才有效

2.不退出Esc方式

5.5 小汇编的使用方法

5.5.1 小汇编的进入和退出

小汇编是CEC-Ⅰ A固化在ROM中的一个工具软件，小汇编的作用是将6502汇编助记符译成6502的机器指令。

在BASIC状态下，用SASM语句可以进入小汇编状态：

```
] SASM
!
```

当屏幕上出现小汇编的提示符“！”之后，表示系统已进入小汇编状态。

当需要退出小汇编状态时，可以键入

Ctrl-C

进入监控状态，也可以键入

Ctrl-Reset

进入BASIC状态。

5.5.2 汇编过程

在小汇编状态下，可按下列格式进行操作：

〈 地址 〉:〈 汇编助记符 〉

当上述格式的一行命令输入后，小汇编程序立即将助记符所对应的机器指令依次放到指定的地址中去，然后，将输入的指令显示在屏幕上，其格式是：

地址－　　机器码　　助记符　　操作数

例

```
! 300: LDA  #$FF
0 300- A9    FF    LDA    #$FF
```

在输入完一条指令后，如果还要继续汇编新的指令，可以用一个空格代替地址，小汇编会自动算出紧接上一条指令的新地址，并完成汇编工作。

例

```
! 300: LDA $1
0300-  A5  01    LDA $01
! ␣ LDX  $2
0302-  A6  02    LDX $02
```

在提示符"!"之后，至少要输入一个空格，否则将出错。当小汇编发现格式错误时，会响铃，同时在出错地方的下面显示一个"^"符号。

如果在提示符"!"之后，紧接着输入一个"$"符号，则在"$"之后可以输入一行监控的命令。

例

```
! $300L
```

表示从$300处反汇编20行指令。

总结小汇编的使用方法，在提示符之后只能紧跟着输入：

① 〈 地址 〉：（在指定地址处开始汇编）

② 空格（从当前地址处开始汇编）

③ $（输入监控命令）

④ Ctrl-C（退出小汇编，进入监控）

第六章　磁盘操作系统的使用

6.1　引言

磁盘操作系统的主要功能是管理磁盘驱动器和管理磁盘文件。在CEC-ⅠA型中华学习机上可以运行APPLEⅡe机上的各种操作系统。如 APPLE DOS、PRODOS、UCSD-P 和 CP/M 操作系统等。本章主要介绍 APPLE DOS 3.3的使用。

6.2　磁盘驱动器

磁盘驱动器是微型计算机系统中一个重要的外部设备。利用磁盘驱动器可以将主机内存的程序和数据记录到磁盘上，并可以从磁盘上读回，达到重复使用、修改的目的。由于配置的磁盘驱动器的类型规格不同，相应的盘控驱动电路、控制接口电路也不完全相同，因此，不同类型的微机系统，其磁盘驱动器并不具有通用性。目前中华学习机 CEC-Ⅰ型、CEC-ⅠA型及APPLEⅡ型的磁盘驱动器是可以通用的。它们的接口电路都支持5.25英寸单面单密度软盘驱动器，并且接口控制信号也互相兼容。

软盘驱动器和磁带录音机一样，都是利用磁化方式来储存数据的。软盘驱动器的存储介质是软盘片，它与磁带之间

的主要差别就在于软盘片如同唱片一样，可以在软盘驱动器中旋转。而软盘驱动器内部的读／写磁头在其定位装置的控制下，可以在磁盘表面的磁道上进行移动，磁头下所经过的磁盘表面存储介质就会被磁化，读／写数据就会记录在磁盘上。

由于磁盘的磁轨由同心圆组成，起始点和终止点相同，这样软盘驱动器就可以随机地读／写软盘上的信息，因此，它要比磁带机的顺序查找、读／写数据的方式快得多，也准确得多。

使用磁盘驱动器时应注意以下几点：

1．使用软盘驱动器要轻拿轻放，避免剧烈震动，以防定位机构松动。

2．避免在软盘驱动器工作时移动驱动器。

3．禁止在开机状态拔插软盘驱动器的连线。

4．不要在马达转动过程中抽、插盘片。

5．避免灰尘的侵袭，不要将沾有灰尘的软盘片插到软盘驱动器中。

6．磁头的清洗，可用专用的清洗软盘片或磁带清洗液。

7．软盘驱动器应远离磁场发生源，不用时不要将软盘片留在驱动器中。

6.3 软盘片

6.3.1 软盘片的包装与外形特点

软盘片密封于永久性保护套内（参见图6-1），盘片由

圆形聚脂塑料做成。盘片表面涂有磁性材料。使用时，软盘片在保护套内旋转，磁头通过保护套中的长孔（又称磁头读／写窗口）和磁盘表面接触，读／写数据。

・永久性保护套：由防静电皮革做成，内壁夹有防静电防潮材料，其作用是防止灰尘和保护盘面磁层不受损伤，同时也防止盘片旋转时所产生的静电而使信息丢失。

・索引孔：用来进行盘片划分扇区的物理定位标志。由于大部分软盘驱动器中没有装上索引孔识别装置，故索引孔不起作用。

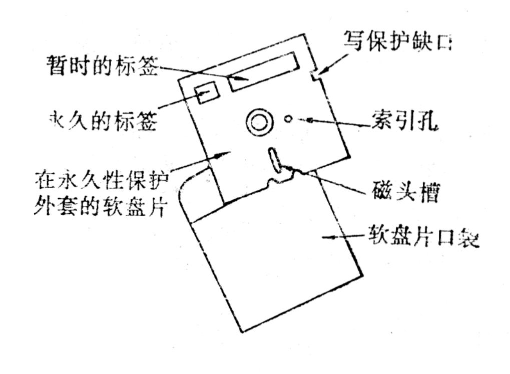


图 6-1

・磁头读／写窗口：这是在永久性保护套上开辟的一个长孔、是磁头与软盘片接触，读／写信息的界面。因此，在软盘片的使用过程中，绝对禁止用手指或异物接触这个长孔中所暴露出的磁盘部分。

・写保护切口：控制着软盘驱动器写电路开关，当用写保护封条贴在该切口上时，可以防止磁盘被写入任何信息。

·永久性标签：出厂时贴上的磁盘说明。用以指明磁盘的牌号和盘片的型号等。在使用盘片时，我们最好用手指捏着这部分取盘或插盘。

·临时标签：在使用软盘时，我们希望能够通过简单的方法注明该盘片的名称、作用等。此时我们可以用不干胶标签贴上，用软笔写上名称、标记等。不需要时，可以撕该标签，换贴新的标签。

·盘纸套：用来保护盘片，防止灰尘的侵蚀。我们要养成把不用的盘片及时插入盘片纸套中的习惯。

6.3.2 软盘片的选择

软盘片有多种型号和规格，它适用于不同类型的软盘驱动器。按照软盘驱动器磁头工作方式可以分成单面和双面读写，按照数据的记录方式又有单密度和双密度二种，按照扇区划分又有硬分段和软分段之分。

对于中华学习机和 APPLE Ⅱ 来说，选用的是5.25英寸单面单密度软盘驱动器。因此，它对软盘片的选择要求比较低。一般地说，单面单密度、单面双密度、双面单密度和双面双密度的5.25英寸软分段软盘片均适合其使用。

6.3.3 磁道和扇区

软盘驱动器在读写信息过程中，由于盘片的旋转，读／写磁头将把盘片划分成若干个同心圆。这些同心圆称为磁道，如图6-2所示。

读／写磁头通过软盘的读／写窗口，可以从一个磁道移

到另一个磁道，这可使磁头找到要读写的数据区。

软盘上的数据是以数据块的形式存放在磁道上的，每一个数据块称为一个扇区。

在 APPLE DOS3.3 操作系统支持下，每张软盘被划分

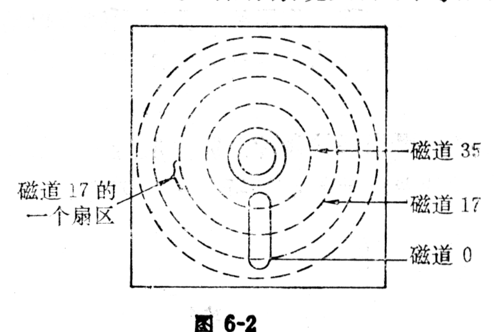


图 6-2

为35个磁道，每个磁道上有16个扇区，每个扇区可存放256个字节的信息（不包括数据块的地址索引）。因此每张盘片的最大存储容量为140KB左右。

6.3.4 软盘片使用注意事项

1. 软盘片不能受重压，不可弯曲，使用后要及时插入盘片的纸盘套内，立放于软盘盒中。

2. 使用时把软盘片有标签的面朝上，用右手拇指按着有永久性标签处，将有磁头读／写窗口的一边先插入驱动器，用拇指慢慢推入软盘片，直到盘片全部进入驱动器。

3. 取出时用食指和拇指捏着盘片的边缘、反方向抽出。

4. 不要在驱动器旋转和亮灯情况下，打开驱动器门，移出软盘片。这样容易损坏软盘片，丢失盘片上的信息。

5. 使用中要注意保持盘片的清洁，避免灰尘和油污的

污染，绝对禁止用手触摸暴露在磁头读／写窗口和索引孔中的磁盘部分。

6．盘片的保存温度最好在10℃～50℃之间，要避开磁场的干扰，保持干燥。

6.4 磁盘操作系统的引导

要使操作系统进行工作，首先遇到的问题是如何将操作系统程序从软盘上装入到主机内存中来，然后才是怎样使用操作系统提供的命令进行操作。这种将操作系统装入内存并使其工作的过程称为磁盘操作系统的引导。

在中华学习机上可以有两种方式来进行磁盘操作系统的引导，即开机引导和命令方式引导。

6.4.1 开机引导

中华学习机固化软件中有一个自启动监控程序，开机后机器首先执行的是这段程序。这个程序会从与主机直接相连接的软盘驱动器（设备号为：S6，D1）中读磁盘，装入磁盘操作系统。因此，只要用户在开机前装好软盘驱动器，并将操作系统盘插入软盘驱动器，然后，打开主机电源开关，机器便会自动装入操作系统。

这种开机引导磁盘操作系统的方式称为“冷启动”方式或“自启动”方式。

6.4.2 命令方式引导

如果主机已加电运行，需要装入或重新装入磁盘操作系

统时，可以将操作系统盘插入软盘驱动器，据据下列所述状态，打入相应命令，即可进行操作系统的引导。

系统状态	提示符	打入命令
APPLESOFT	]	PR#6或IN#6
整数 BASIC	>	PR$6或IN#6
监 控 程 序	*	6Ctrl-P或C600G〔注〕
小 汇 编	!	$ C600G

这种在主机已加电情况下，引导操作系统的方法又称为“热启动”。

6.5 DOS 命令格式

DOS 操作系统命令由命令保留字和命令参数二部分组成。

命令保留字代表了 DOS 操作系统所提供的命令。它必须由大写字母组成。

命令参数则是提供 DOS 命令执行时所需要的辅助信息。如：磁盘驱动器的设备号，文件存放的地址，磁盘的卷号，文件名等等。命令参数由文件名、参数标记和参数组成。参数标记必须是大写字母，参数则可以是算术表达式。

一条DOS命令可以带有若干命令参数，命令参数之间必须用逗号隔开。

DOS 命令格式如下：

〔注〕“Ctrl-P”表示在键盘上按下Ctrl键的同时，按下P键。“Ctrl-A”等类似。

```
COMMAND f  [, Ss]   [, Dd]   [, Vv]
   |        \_____________________/
命令保留字          命令参数
```

在上述DOS命令语法描述中，方括号（[和]）之中的项为可选择命令参数，大写字母为参数标记，小写字母为参数。可选择命令参数的顺序是任意的。当命令所选用的有关驱动器所在槽号、设备号省略时，总是指定最后使用操作的磁盘设备为当前DOS命令所执行的设备。

命令参数说明：

1. f 文件名，以字母打头的字符串组成（其中不能含有逗号），文件名的有效长度最多可达30个ASCII字符。

2. Ss 设定软盘驱动器所在的槽口号，s 值可以是1～7之间的整数（但不允许为3）。

中华学习机盘驱接口上所接盘驱的槽口号 s＝6。

3. Dd 设定软盘驱动器在盘驱接口卡上的设备号。d 值必须是1或2。接口上所接盘驱的设备号 d＝1。

4. Vv 软盘片的卷号，一张软盘片只能有一个卷号，卷号的值 v 可以是 1—254 之间的整数。

卷号是在用 INIT 命令进行盘片初始化操作时就已确定了的，是不可更改的，卷号起着核对盘片的作用。

5. Bb 用以指明文本文件起始字符的位置，b 值可以是0～32767之间的整数。

6. Rr 在顺序文件中用来指明数据域起始标志，而在随机文件中则用来指明记录数。r 值可以是0～32767之间的整数。

6.6 DOS 命令执行方式

在装入 DOS 操作系统后，便可以使用 DOS 的命令。DOS的命令有两种执行方式，即立即方式和程序方式

6.9.1 立即执行方式

在 BASIC 提示符下，直接打入 DOS 命令，DOS 将立即解释执行该命令。这种工作方式类似于 BASIC 中的立即型命令。

可以立即执行的 DOS 命令有：

BLOAD	BRUN	BSAVE	CATALOG	CLOSE
DELETE	EXEC	FP	INIT	IN#
INT	LOAD	LOCK	MAXFILES	MON
NOMON	PR#	RENAME	RUN	SAVE
UNLOCK	VERIFY			

6.6.2 程序调用方式

在 BASIC 程序中，DOS 操作系统命令均可作为一条特殊的BASIC 语句被调用。它与一般 BASIC 语句不同的是：DOS 操作系统命令必须通过一条特殊的 PRINT 语句来执行。这条PRINT语句将DOS 命令及其命令参数作为一输出行进行输出，而在这输出行的第一个字符是 Ctrl-D 。

Ctrl-D 是 DOS 命令的引导标记，当 BASIC 解释程

序执行 PRINT 语句时，遇到第一个输出字符为 Ctrl-D，则作为 DOS 命令去处理，它不会将Ctrl-D字符及其后面的 DOS 命令字符串作为一般的输出字符显示到屏幕上或输出设备上（当然，如果系统中没有启动 DOS 操作系统，BASIC 程序是不会对 Ctrl-D 字符作特殊处理的）。

调用 DOS 命令的语句格式：

PRINT ″Ctrl-D DOS命令和命令参数″

或PRINT CHR $（4）；″DOS命令和命令参数″

对于整数 BASIC 程序，由于没有 CHR $（4）函数，所以只能用第一种语句格式，对于 APPLESOFT 程序，二种语句格式均可以使用。

如果我们把 Ctrl-D 字符先赋值给串变量，上述语句可改写为：

D $ = ″Ctrl-D″或D $ = CHR $(4)

PRINT D $ ″DOS 命令和命令参数″

下面这段程序将显示盘片上的文件目录：

```
10  REM DOS 3.3 COMMAND
20  PRINT″CATALOG OF DOS 3.3 SYSTEM
    MASTER″
30  PRINT CHR $ (4) ; ″CATALOG″
40  END
```

在使用 BASIC 语句调用 DOS 命令时应注意：

(1) 在一个PRINT语句输出行中，只能含有一条 DOS 命令，而在这个输出行中必须以 Ctrl-D 字符打头，以回车字符（Ctrl-M）表示 DOS 命令的结束。

请比较一下下面二段程序为什么不执行 DOS 命令和出现语法错的原因：

例 1

```
10  REM DOS 3.3 COMMAND
20  PRINT" CATALOG OF DOS 3.3
    SYSTEM MASTER";
30  PRINT CHR $ (4) ; "CATALOG"
40  END
```

例 2

```
10  REM DOS 3.3 COMMAND
20  PRINT CHR $ (4) ; "CATALOG"
30  PRINT CHR $ (4) ; "CATALOG"
40  END
```

仅能以程序调用方式执行的 DOS 命令有：

APPEND　　CLOSE　　OPEN　　POSITION

READ　　WRITE

6.7 DOS命令的使用

6.7.1 格式化新盘片（INIT）

INIT 命令为格式化命令，更确切地说叫初始化命令，因为此命令不仅对盘片进行格式化，而且还将操作系统写入盘片的 \$00～ \$02三个磁道上，并把当前内存中的BASIC程序存入磁盘作为操作系统引导结束后自动执行的程序。

命令格式：INIT f [, Ss][, Dd][, Vv]

INIT 命令一般仅用于对初次使用的空白盘片进行初始化，但当某张已使用过的软盘片上所存储的文件已全部无用，并打算将该盘片用来重新存放一些新文件时，亦可对其进行初始化操作。执行初始化操作后，原盘片上的所有信息都将被清除。

INIT 命令将对软盘执行以下操作：

(1) 对盘片格式化，删除盘片上所有信息，在磁盘表面划分磁道和扇区段。并将所有区段内容写成0。

(2) 将 DOS 操作系统存放到软盘的 $00～ $02三个磁道上。

所有经过初始化的软盘上均装有 DOS 操作系统，也均可以作为 DOS 系统的引导盘。

(3) 在 $11道上为该盘片建立文件目录区，填写该盘所属卷号。

卷号可以通过 Vv 参数由用户自行指定，当 Vv 参数省略，或 v=0 时，将自动分配一个卷号为 254，一张盘片只能有一个卷号，一旦建立就不能被修改。

(4) 将当前已建立在主机内存中的 BASIC 程序作为引导 DOS 操作系统后，自动装入执行的第一个程序（通常称为问候指序）存放到软盘上，冠以指定的文件名f。

(5) 用 INIT 命令初始化后的软盘，只能是"从属盘"，它不能像 DOS 3.3 系统主盘一样可以在 16～64KB RAM 的任一类型的 APPLE Ⅱ 机上进行启动，但可以通过 MASTER CREATE 程序的运行使其变为系统主盘（详见第6.8.8 节）。

例如，下面这些步骤的操作将初始化一张空白盘片：

(1) 用DOS 3.3系统盘启动。

(2) 清除内存，打入如下程序：

```
10  PRINT "THIS DISK BELONG TO
  ZHANG MING"
```

(3) 打入命令RUN检查运行结果。

```
] RUN
  THIS DISK BELONG TO ZHANG MING
```

(4) 取出系统主盘，插入空白盘。

(5) 打入初始化命令：

```
] INIT HELLO
```

(6) 从盘驱中取出软盘片，贴上临时标签，不用时，及时插入磁盘的纸套中并放入盘片盒中。

6.7.2 管理磁盘文件的DOS命令

1. 列文件目录（CATALOG）

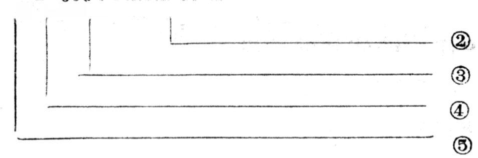
用途：列磁盘文件目录。

格式：CATALOG[, Ss][, Dd][, Vv]

说明：该命令将按命令所设定的软盘驱动器列出该盘驱上盘片的文件目录。目录中包括：磁盘的卷号，文件的名称，文件所占用盘片的扇区数，文件的类型和文件的封锁标志等等。例如：

```
] CATALOG
DISK VOLUME254——————————①
* A 008 HELLO
* I 018 ANIMALS
  T 003 APPLE  PROMS
* I 006 APPLESOFT
```

```
* I 026 APPLEVISION
* I 017 BIORHYTHM
* B 010 BOOT13
* A 006 BRIAN'S THEME
* B 003 CHAIN
* I 009 COLOR DEMO
* A 009 COLOR DEMOSOFT
* I 009 COPY
* B 003 COPY, OBJ0
* A 009 COPYA
* A 010 EXEC DEMO
* B 020 FID
* B 050 FPBASIC
* B 050 INTBASIC
```



通过屏幕上所列出的这些信息，我们可以知道以下几件事情：

① 该盘片所占有的卷号。一张盘片只能有一个卷号，它是在初始化命令执行过程中所确定的。

② 文件名，每个文件名最大长度为30个字符。

③ 文件在软盘上所占用的扇区数。

④ 文件类型：

A 表示APPLESOFT程序文件。

I 表示整数 BASIC 程序文件。

B 表示2进制代码文件。

T　表示文本文件。

R　表示由 EDASM 汇编程序所产生的可浮动目标文件。

L　表示由 LISA 汇编程序所生成的文件。

⑤　文件保护标志。它是操作系统所提供的对文件进行写保护的一项措施。

如果该文件有星号标志(*)，则表示 DOS 对该文件进行了封锁保护，它只允许用户对该文件执行读操作，而不能对其执行改名、删除或写操作。否则，系统会给出：FILE LOCKED 信息。

在列目录过程中，可以用 Ctrl-S 键暂停列目录，再按任意键使其继续列目录。如果目录内容多于幕屏一次能够显示的条目时，可按任意键使其继续显示。

2. 文件加锁 (LOCK)

用途：置文件封锁。

格式：LOCK f[, Ss][, Dd][, Vv]

说明：该命令将对指定的文件置封锁（写保护）状态，使得该文件不会因意外的写操作而遭破坏。

对某一文件执行封锁操作后，列磁盘文件目录，该文件类型标志的左边将有一星号标志(*)，对于被封锁的文件，用户只能执行LOAD／READ／EXEC／VERIFY操作，而不能对其执行 SAVE／WRITE／RENAME／DELETE 等操作。否则将会得到信息：FILELOCKED。

例如，插入刚刚初始化完毕的工作盘，列出磁盘文件目录

```
]CATALOG
```

```
DISK VOLUME 010
A 002 HELLO
```

现在执行LOCK HELLO 命令，再次列出磁盘文件目录时，HELLO 文件将被封锁。

```
]LOCK HELLO
]CATALOG
DISK VOLUME 010
* A 002 HELLO
```

8. 文件解锁（UNLOCK）

用途：解除文件封锁。

格式：UNLOCK f [, Ss][, Dd][, Vv]

说明：该命令将解除对文件的封锁，这样可以使得对软盘上的该文件进行删除、重新命名、修改或重写等操作，解除文件封锁后，该文件的封锁标志将消失。

LOCK 和 UNLOCK 命令可以对任意类型的文件进行操作。

4. 修改文件名（RENAME）

用途：修改文件名。

格式：RENAME f1 f2 [, Ss][, Dd][, Vv]

说明：该命令将对指定的文件重新命名。用文件名 f 2 取代原有的文件名 f 1，改名后的文件其内容不受影响，可以对任意类型文件执行该操作。

这里的 f1 是一个已在软盘上存在的文件名，f 2 必须是软盘上所没有的新文件名，f 1 文件必须不被封锁。

例：若工作盘上的文件目录为：

```
]CATALOG
DISK VOLUME 010
```

* A 002 HELLO

A 009 ABC

现欲将 ABC 文件改名为BCD，可执行如下命令：

```
] RENAME ABC, BCD
] CATALOG
DISK VOLUME 010
* A 002 HELLO
  A 009 BCD
```

注意：RENAME命令并不检查文件名f 2是否在软盘上已存在，因此有可能通过 RENAME 命令而使得在一张软盘上有多个同名文件，最好要避免这种不必要的混乱出现。

5．文件校验（VERIFY）

用途：文件校验。

格式：VERIFY f [, Ss][, Dd][, Vv]

说明：该命令将通过读操作对指定文件f的校验和进行检查，以确定该文件在软盘上的信息是否正确。

我们知道，DOS 操作系统在把信息存到软盘上的过程中，首先把信息存入文件缓冲区，待存满 256 个字节时，DOS 再把这些信息作为一个数据块存到软盘的扇区中。这时，存到该扇区中的信息除了 256 个字节是文件信息外，还有一个作为“检查和”的校验字节，据据这个校验字节，可以检查文件传输的正确性。

一般说来，从主机写到软盘上的信息都是很可靠的，通过 DOS 命令的使用是可以重新将它读回到内存中来。但是由于人为因素，盘片长期存放，软盘片部分地遭到损伤，使得信息被丢失，DOS 不得将它读回到内存中来。通过 VERIFY 命令可以对盘上的文件进行一次读操作，校验一

下该盘片上文件的正确性。

VERIFY 命令操作过程中，如果被校验的文件正确，将得不到任何回答信息，否则，将给出I/O ERROR信息。这将提示你，该文件已遭破坏，这时应考虑重新复制该文件，或者将该盘上的其它文件盘制到其它软盘上，以避免整张盘片被损坏。DOS3.3 操作系统在存储每个文件时，都将自动执行一次 VERIFY 操作，以保证存盘操作是成功的。

VERIFY 命令可适用于对任何类型的文件进行校验操作，该操作过程将不破坏主机内存中程序和数据，也不影响软盘上的文件。

6. 删除文件（DELETE）

用途：删除文件

格式：DELETE f [, Ss][, Dd][, Vv]

说明：该命令将从软盘上删除所指定的文件，这个文件可以是任意类型的，并且文件是不被封锁的。

如果要删除的文件不存在，将出现信息：FILE NOT FOUND。如果该文件被封锁，将得到信息：FILE LOCKED

6.7.3 与BASIC程序有关的DOS命令

1. 装入BASIC程序（LOAD）

用途：装入 BASIC 程序。

格式：LOAD f [, Ss][, Dd][, Vv]

说明：该命令将从软盘上装入一个类型为 A 或 I 的 BASIC程序。

如果命令中所指出的文件名不在软盘上，将得到信息：FILE NOT FOUND。如果欲装入的文件，其文件类型不

是 A 或1，则会得到信息：FILETYPE MISMATCH。如果没有用DOS3.3系统主盘进行启动，或没有将整数BASIC解释程序装入到主机内存中，而要装入的却是 I 类型文件，将会得到信息：LANGUAGE NOT AVAILABEL。

LOAD 命令装入 BASIC 程序之前，首先要检查磁盘文件的类型，据据装入的是 A 类型文件还是 I 类型文件，转入 APPLESOFT 解释程序或整数 BASIC 解释程序，然后才将该文件的内容装入到主机内存中。

在装入 BASIC 程序时，盘上的文件保持不变，原先已在内存中的 BASIC 程序将被清除。

2. 保存BASIC程序（SAVE）

用途：保存 BASIC 程序。

格式：SAVE f[，Ss][，Dd][，Vv]

说明：该命令将把当前已建立在主机内存中的 BASIC 程序保存到软盘上，以建立程序副本。

f 指出的是程序副本的文件名。文件类型根据当前系统处于何种 BASIC 语言状态所确定。如果该盘上没有文该件名，则在软盘上创建该文件。如果该文件名在软盘上已存在，并且文件类型与系统的当前状态相吻合，则用当前已在内存中的 BASIC 程序替代原有文件。如果软盘上的文件类型和主存中 BASIC 程序类型不一致，则给出信息：FILE TYPE MISMATCH。

利用SAVE命令，我们可以在一个软盘上建立多个程序副本，也可以在多个软盘上建立多个程序副本。

例如，我们把在 DOS3.3 系统主盘上的 COLORDEMO 程序存到工作盘上，可以执行如下操作：

(1) 用 DOS 3.3 系统主盘进行启动。

(2) 打入命令：LOAD COLOR DEMO

(3) 换上工作盘，打入命令：SAVE COLOR DEMO

这样，在你的工作盘上，便保存了一个和DOS3.3 系统主盘上一模一样的整数 BASIC 程序：COLOR DEMO。

3. 运行BASIC程序（RUN）

用途：运行软盘上的BASIC程序。

格式：RUN f[，Ss][，Dd][，Vv]

说明：该命令将把软盘上的某一BASIC程序f装入到内存并运行之。

该命令的执行步骤是：①执行 DOS 的 LOAD 命令，②执行 BASIC 的 RUN 命令。

6.7.4 语言之间进行转换的DOS命令

DOS 3.3 操作系统为方便用户的使用提供了 CEC-BASIC 解释程序和整数 BASIC 解释程序二种语言状态相互切换的命令。

1. 进入整数 BASIC（INT）

用途：进入整数 BASIC。

格式：INT

说明：该命令使系统进入整数BASIC状态，并清除内存中现有程序，恢复有关程序指针。

如果我们想从 CEC-BASIC 状态进入整数 BASIC状态，可以使用这一命令。该命令不带任何参数，它不能设定软盘驱动器的槽号和设备号。

如果机内没有装入整数 BASIC 解释程序，系统将给出

信息：LANGUAGE NOT ANAILABEL。

2. 进入浮点BASIC（FP）

用途：进入 CEC-BASIC。

格式：FP[，Ss][，Dd]

说明：CEC-BASIC 程序又称为浮点 BASIC 程序，该命令将使得系统进入 CEC-BASIC 状态。并清除内存中现有的BASIC程序，恢复各有关指针。

如果我们在运行了某一整数 BASIC 程序，想回到CEC-BASIC状态，可以打入FP命令，如果我们想清除当前内存中的 BASIC 程序，可以打入 FP 命令。如果我们因运行了某个 BASIC 程序，使得系统有关指针的改变，现在想装入第二个程序执行时，系统回答"OUT OF MEMORY"信息拒绝执行时，可以使用FP命令。

6.7.5 对文本文件进行操作的DOS命令

文本文件是用来存放数据信息的文件，其数据信息可以是：运算数据、计算的结果、信件的副本、帐单和表格等，也可以是组成DOS命令和BASIC 命令的字符串。

文本文件是通过程序方式调用 DOS 命令来创建和检索的，它可以通过某一程序来创建，而由另一程序进行检索。

文本文件按其存取方式可以分成顺序文件和随机文件。这两种文件在目录中的类型标识符均为 T、但在使用方法和存储格式上却有所不同，如果使用不当，可使得存取过程中所得到的数据结果与预期的大不相同，甚至会得到出错信息。

1. 顺序文件

顺序文件由数据域组成，每个数据域由回车（CR）字符作为终结标志，它通常由不带逗号和分号的PRINT语句自动生成，也可以由输出CHR$（13）字符来产生。顺序文件在软盘上是连续存放的，各个数据域之间没有间隙。其数据域的长度和文件长度可以是任意的。

顺序文件在盘上的存放格式如图6-3所示：

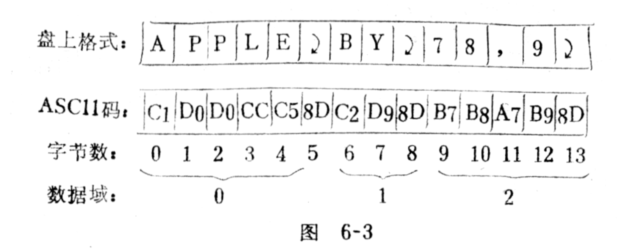


图 6-3

（1）顺序文件的一个例子

假设我们想把单词：APPLE、BANANA、FRUIT、EAGLE 和 GOOSE 作为一个文本文件保存到软盘上，我们可以用如下的程序来完成：

```
10  REM MAKE-WORD1
15  DIM A $ (10)
20  D $ =CHR $ (4) : REM CTRL-D
30  PRINT D $; "OPEN WORD1"
40  PRINT D $; "WRITE WORD1"
50  FOR I=1 TO 5: READ A $ (I) :PRINT
    A $ (I) : NEXT I
60  PRINT D $; "CLOSE WORD1"
70  END
100 DATA"APPLE", "BANANA", "EAGLE"
    "GOOSE", "FRUIT"
```

在这个程序中，我们使用了三条DOS命令：OPEN、WRITE和CLOSE，其作用是：OPEN创建一文本文件WORD1，并把该文件建立在目录中，WRITE命令对WORD1文件执行写操作，使得第50列语句中的PRINT命令将把输出信息送到文件WORD1中，而不是屏幕上，CLOSE命令则是停止对WORD1文件执行写操作。

如果我们不是在DOS的监视命令方式下运行该程序，屏幕上将看不到任何信息，如果我们打入置监视方式命令：MON C，I，O，然后再打入RUN命令，则会在屏幕上看到如下信息：

```
] MON C, I, O
] RUN
OPEN WORD1
WRITE WORD1
APPLE
BANANA
EAGLE
GOOSE
FRUIT
CLOSE WORD1
```

这样我们可以确信已把这些单词保存到了WORD1文件中，那么如何再从这个文件中读回这些信息呢？下面这个程序将帮助你解决这个问题：

```
] LIST
10  REM RETRIEVE-WORD1
15  DIM B $ (10)
20  D $= CHR $ (4) : REM CTRL-D
30  PRINT D $; "OPEN  WORD1"
```

```
40  PRINT D $; "READ WORD1"
50  FOR I =1 TO 5 :INPUT B $ (I) :NEXT I
60  PRINT D $; "CLOSE WORD1"
70  END
```

在这个程序中，我们同样也使用了三条 DOS 命令：OPEN、READ 和 CLOSE。这里的 OPEN 命令将打开 WORD 1 文件，READ 命令将告诉 DOS：第 50 行语句中的 INPUT 命令将从 WORD1 文件中读取数据，而不是从键盘上。CLOSE 命令将关闭 WORD 1 文件，结束对 WORD 1 文件执行读操作。

如果打入 MON C，I，O命令，再运行该程序，我们可以看到如下信息：

```
] MON C, I, O
] RUN
OPEN WORD 1
READ WORD 1
? APPLE
? BANANA
? EAGLE
? GOOSE
? FRUIT
CLOSE WORD 1
```

为了进一步确认WORD1文件中的数据被读了回来，我们可打入如下命令：

```
] PRINT B $ (1) , B$ (3) , B$ (5)
APPLE          EAGLE          FRUIT
```

通过这二个例子我们看到了顺序文件的写过程和读过程，与它所使用的DOS命令有着密切的关系。

(2) OPEN

用途：打开顺序文件。

格式：OPEN f [, Ss][, Dd][, Vv]

说明：该命令将打开一顺序文件f，如果该文件不存在，则在磁盘的目录中创建该文件。

在对某一文件进行读操作或写操作之前，首先必须打开该文件。此时 DOS 将给该文件分配595个字节的文件缓冲区，供磁盘与主存之间的信息交换用。程序一次允许同时打开的文件个数是由 MAXFILES 命令所决定，最多不得超过16个，DOS 启动后将自动执行 MAXFILES3 命令，此时系统允许打开的文件个数最多为3个。

(3) CLOSE

用途：关闭文件。

格式：CLOSE [f]

说明：对某一文本文件在完成读／写操作后必须及时关闭该文件，使得 DOS 命令收回分配给该文件的文件缓冲区，使它可以用于新打开的文件。否则会由于打开的文件太多，而出现：NO BUFFER AVAILABLE 信息。

该命令执行时，f指出了要关闭的文件名，当文件名省略时，将关闭所有已打开的文件。

(4) WRITE

用途：写顺序文件。

格式：WRITE f [, Bb]

说明：该命令将对已打开的文件 f 执行写操作，参数 Bb 指明从顺序文件的第 b 个字节开始写数据，如果参数 Bb 省略则从顺序文件的第 0 个字节开始写数据。执行该命

令后，所有 PRINT 语句输出的信息都将存到文件中，而不是显示到屏幕上。

注意：如果打开了盘上已有的文本文件，再对它进行执行写操作，此时除非新写入的数据多于原有文件的信息，并覆盖了原有的信息，否则得到的文件内容将是新写入的信息和原来文件混合构成。

在下面这个例子中，我们虽然在WORD1文件中只存储了二个数据，但因原来已有5个数据，所以从该文件中读到了5个数据。

```
10  REM MAKE WORD1
15  DIM  A $ (10) , B $ (10)
20  D $ =CHR $ (4) : REM CTRL-D
30  PRINT D $; "OPEN WORD1"
40  PRINT D $; "WRITE WORD1"
50  FOR  I=1 TO 2:READ A $ (I) :PRINT
    A $ (I) : NEXTI
60  PRINT D $; "CLOSE WORD1"
80  DATA 123, 456
100 REM RETRIVE WORD1
110 PRINT D $; "OPEN WORD1"
120 PRINT D $; "READ WORD1"
130 FOR I=1 TO 5: INPUT B $ (I) :NEXT I
140 PRINT D $; "CLOSE WORD1"
150 END
```

打入MON C，I，O 命令后，打RUN 命令所以从屏幕上得到如下信息：

```
] MON C, I, O
] RUN
OPEN WORD1
```

```
WRITE WORD1
123
456
CLOSE WORD1
OPEN WORD1
READ WORD1
? 123
? 456
? NANA
? EAGLE
? GOOSE
CLOSE WORD1
```

通过这个例子，我们知道 WORD1 文件的内容是由二个程序的写入数据混合构成的，为了避免出现这种混乱，使得到的新文件内容只有新写入的数据，可以采用打开—删除—打开的方式来建立新文件。在该例中的30行语句前面增加二条语句：

```
22  PRINT D $; "OPEN WORD1"
24  PRINT D $; "DELETE WORD1"
```

这样无论磁盘上是否有WORD1文件，都不会出现混乱情况。

(5) READ

用途：读顺序文件。

格式：READ f [, Bb]

说明：该命令将对已打开的文件 f 执行读操作，参数Bb指明了从顺序文件的第b个字节开始读数据，如果参数Bb省略则从顺序文件的第 0 个字节开始读数据。执行该命令后，所有INPUT语句或GET语句都将从文件中读到数据，

而不是从键盘上读数据。

注意：在用 INPUT 语句读入数据时，一次读入一个域，并把这个域中以逗号分隔的数据串依次赋给 INPUT语句中的各个变量。

如果变量的类型和所给的数据类型不匹配，则给出？REENTER 信息，继续取下一个数据，如果数据域中的数据多于 INPUT 语句中的变量个数，则给出信息：？EXTRA IGNORED，忽略多余的数据部分。如果数据域中的数据个数少于INPUT语句中变量的个数时，屏幕将出现？？信息，自动到下一个数据域中取数据。这种情况的出现和BASIC语句一样，是我们所不希望的错误，会使得读数据丢失。

防止读数据出错的有效办法是：检索程序读数据的格式应按照写数据文件的格式来设计，变量类型也应和写数据程序的变量类型相一致。

(6) APPEND

用途：使文件添加内容。

格式：APPEND f [, Ss][, Dd][, Vv]

说明：该命令将自动打开顺序文件 f，并把文件位置指针移到文件的末尾（即文件最后一个字符的后面）。很明显，使用该命令的主要目的是在文件的末尾增加新的信息。

在 APPEND 命令之后必须使用 WRITE 命令，如果使用了 READ 命令，执行 INPUT 语句时，则会显示信息：END OF DATA。

下面这个例子将在 WORDI 文件中增加 2 个单词，SHEEP和WOOD。

```
10 REM APPEND-WODR1
```

```
20  D $ =CHR $ (4)
30  PRINT D $; "APPEND WORD1"
40  PRINT D $; "WRITE WORD1"
50  PRINT "SHEEP"
60  PRINT "WOOD"
70  PRINT D $; "CLOSE WORD1"
80  END
```

相应的检索程序可改为：

```
10  REM RETRIEVE-WORD1
15  DIM B $ (10)
20  D $ = CHR $ (4)
30  PRINT D $; "OPEN WORD1"
40  PRINT D $; "READ WORD1"
50  FOR I=1 TO 7: INPUT B $ (I) :NEXT I
70  PRINT D$; "CLOSE WORD1"
80  END
```

(7) POSITION

用途：文件的定位。

格式：POSITION f [, Rr]

说明：该命令用来对顺序文件执行定位操作，它可以把文件位置指针从当前位置移到下面第 r 个域的起始位置。参数Rr 指明的是当前位置的第 r 个域。当参数Rr省略时，表示不移动文件的位置指针。

POSITION 命令必须作用于一个已打开的顺序文件，文件名 f 必须与 OPEN 命令中的文件名一致，它和其它 DOS 命令一样，将撤消 WRITE 和 READ 命令的功能，所以，POSITION 命令必须用在 WRITE 和 READ 命令之前，否则，使用过 POSITION 命令之后还需要重新使用

READ 或 WRITE 命令，才能执行读操作或写操作。

例：在 MAKE-WORD1 例子程序中，我们已建立了一个顺序文件WORD1，它由五个域构成，每一个域包含了一个单词。APPLE 单词在文件中是第 0 域。现在我们想从 WORD1文件中读取第三、四两个域中的单词，可用下面程序来实现。

```
10 REM POSITION-WORD1
20 D $ = CHR $ (4)
30 PRINT D $; "OPEN WORD1"
40 PRINT D $; "POSITION WORD1, R3"
50 PRINT D $; "READ WORD1"
60 INPUT A $
65 INPUT B $
70 PRINT D $; "CLOSE WORD1"
80 PRINT A $: PRINT B $
90 END
```

打入 RUN 命令可得到：

```
] RUN
GOOSE
FRUIT
```

2. 随机文件

随机文件由记录组成，它以记录为单位进行数据的读操作和写操作。每个记录中可以包含有许多数据域，各个记录的记录长度是固定的，一个记录的最大长度是32767个字符。

设有一随机文件，它由三个记录构成，记录长度为6，它在盘上的存放格式为图6-4。

从这里可以看出顺序文件是随机文件的一个特例，其中所含的记录个数为1。

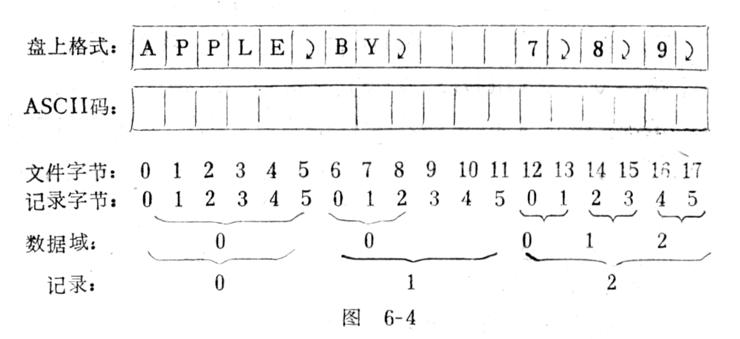


图 6-4

使用随机文件，可以实现快速地查找、增补或修改其中任一记录中的数据，而不影响其它记录中的内容。因此，象帐目管理一类的应用软件特别适用于随机文件。

(1) OPEN

用途：打开随机文件。

格式：OPEN f，Lj[，Ss][，Dd][，Vv]

说明：该命令将打开一随机文件f，其记录长度由参数 Lj 来决定，j是1～32767之间的数。如果文件 f 不存在，则在磁盘目录中建立该文件。

在打开随机文件时，DOS 并不把该文件的记录长度存入磁盘，如果我们重新打开 f 文件时，参数 Lj 所指明的长度与创建时指明的长度不一致，则 DOS 将按照新给的参数来计算文件中记录的位置，这样读写随机文件时可能会造成混乱的结果。为了避免出现这种情况，我们必须保存好关于随机文件的结构和数据格式，以便在编制检索程序时不至于搞错。最为简单的办法是把这些信息作为文件的第 0 个记录保存起来，而把记录长度附在文件名中。例如：MAILER文件可取名为MAILER。L30或MAILER(L30) 等。

(2) CLOSE

该命令的用途和使用格式与3.7.1节CLOSE完全相同。

(3) WRITE

用途：写随机文件。

格式：WRITE f[，Rr][，Bb]

说明：该命令将对随机文件 f 执行写操作。这里有二个参数 Rr 和 Bb，分别用来指明是对随机文件中第 r 个记录执行写操作，其起始位置为b字节。

参数Rr指定文件的r个记录，r 值可以是0～32767之间的一个数，当 r＝0 或省略该参数时表示对 f 文件的第 0 个记录执行写操作。

参数 Bb 表示从该记录中第 b 个字节开始执行写操作。如果b值很大，超出了记录长度，则在使用 PRINT 语句后，写入的数据可能会破坏下一个记录中的内容。

在使用 WRITE 命令后，不能使用其它 DOS 命令或INPUT语句或GET语句，否则将会终止WRITE命令的执行。

(4) READ

用途：读随机文件。

格式：READf [，Rr][，Bb]

说明：该命令将对随机文件 f 执行读操作。这里有二个参数 Rr 和 Bb，分别用来指明是对随机文件中第 r 个记录执行读操作，其起始位置为b字节。

在使用READ命令后，不能使用其它DOS命令PRINT语句，否则将会终止READ命令的执行。

(5) 随机文件的一个例子

这里给出一个创建和使用随机文件的例子。在这个例子中，随机文件的记录长度为30个字符，每个记录中包含有3个域，即姓名N $，电话号码P $和邮政编码Z $。

```
10  REM MAKE MAILER
15  DIM N$ (10) , P$ (10) , Z$ (10)
20  D$=CHR $ (4)
22  PRINT D$; "OPEN MAILER, L30"
25  PRINT D$; "DELETE MAILER"
30  PRINT D$; "OPEN MAILER, L30"
40  FOR I= 1 TO 5
50  PRINT D $; "ERITE MAILER, R"; I
60  READ N $ (I) , P$ (I) , Z $ (I)
70  PRINT N $ (I) :PRINT P$ (I) :PRINT
      Z $ (I)
80  NEXT I
90  PRINT D $; "CLOSE MAILER"
100  DATA ZHANG, 28—2451, 100010
102  DATA WANG, 66—8841, 100012
104  DATA ZHOU, 5—2300, 210031
106  DATA ZEN, 1—7001, 100015
107  DATA XIU, 5—4321, 210033
110  END
```

在内存中建立完该程序后，打入RUN命令，即在MAILER文件中建立含有5个记录的随机文件。

下面这段程序，将把MAILE文件的记录取出来，并显示到屏幕上。

```
10  REM RETRIEVE MAILER
15  DIM N$ (10) , P$ (10) , Z$ (10)
20  D$ = CHR $ (4)
```

```
30   PRINT D$; "OPEN MAILER, L30"
40   FOR I = 1 TO 5
50   PRINT D $; "READ MAILER, R"; I
60   INPUT N $ (I) :INPUT P $ (I) :INPUT
        Z$ (I)
70   NEXT I
80   PRINT D $; "CLOSE MAILER"
90   FOR I = 1 TO 5: PRINT N $ (I) , P$
          (I) , Z$ (I) : NEXT I
100  END
```

打入RUN命令后，我们将得到如下信息：

```
]RUN
ZHANG        28—2451        100010
WANG         66—8841        100012
ZHOU         5—2300         210031
ZEN          1—7001         100015
XIU          5—4321         210033
```

3. 执行顺序文件（EXEC）

用途：执行顺序文件。

格式：EXEC f [,Rr][,Ss][,Dd][,Vv]

说明：该命令将从顺序文件f中第r个域开始执行该文件中所含的命令。这些命令就象是从键盘上一句一句打入的一样。

在这个命令中，参数Rr 指出文件f的一个绝对域址，如果参数Rr省略则表示从文件的开头去执行。

顺序文件f中所含的命令可以是BASIC命令、监控命令和DOS命令。下面这个例子中将建立一个含有DOS 命令、BASIC命令和监控命令的文本文件TEST。

```
10   REM MAKE-EXEC
20   D$=CHR $ (4)
30   PRINT D $; "OPEN TEST"
40   PRINT D $; "WRITE TEST"
50   PRINT"CATALOG"
60   PRINT"CALL-151"
70   PRINT"FD18L"
80   PRINT"FP"
90   PRINT"? 34+16"
100  PRINT D $; "CLOSE TEST"
110  END
```

打入这段程序后，打入命令RUN。此时，在软盘上建立了可执行的顺序文件TEST。

打入命令EXEC TEST，幕屏上将出现信息：

```
]EXEC TEST
]
DISK VOLUME 254
A 002 HELLO
A 002 MAKE-WORD1
T 002 WORD1
A 002 RE-WORDI
A 003 MAK-RE
A 002 MAKE-EXEC
T 002 TEST
]
*
FD18-    6C   38   00      JMP    ($0038)
FD1B-    E6   4E           INC    $ 4E
FD1D-    D0   02           BNE    $ FD21
FD1F-    E6   4F           INC    $ 4F
```

```
FD21—   2C   00   C0    BIT   $ C000
FD24—   10   F5         BPL   $ FD1B
FD26—   91   28         STA   ($ 28), Y
FD28—   AD   00   C0    LDA   $ C000
FD2B—   2C   10   C0    BIT   $ C010
FD2E—   60              RTS
FD2F—   20   0C   FD    JSR   $ FD0C
FD32—   20   A5   FB    JSR   $ FBA5
FD35—   20   0C   FD    JSR   $ FD0C
FD38—   C9   9B         CMP   # $ 9B
FD3A—   F0   F3         BEQ   $ FD2F
FD3C—   60              RTS
FD3D—   A5   32         LDA   $ 32
FD3F—   48              PHA
FD40—   A9   FF         LDA   # $ FF
FD42—   85   32         STA   $ 32
*
]
50
]
```

在使用EXEC命令时要注意以下几点：

（1）如果在f文件中含有EXEC命令，假设为"EXEC g"，那么当执行到EXEC g命令时，将自动关闭f文件，而打开并执行g文件。

（2）如果在f文件中含有DOS的RUN命令，假设为"RUN g"，那么当执行到RUN g命令时，系统将执行RUN g操作，等g程序执行完毕后，再继续执行f文件中的后续操作命令。但要注意，在EXEC命令下运行程序g时，程序中的任何INPUT语句都会将文件f的下一个域作为输入

数据，而不从键盘上输入数据。

(3) 如果在文件f中含有带行号的BASIC程序行，EXEC命令并不执行这些程序行中的命令，而是按其行号次序将它们装入内存中，就象这些BASIC语句行是从键盘上输入到内存中一样。

利用EXEC的这一特性，我们可以把某一子程序插入到主程序中，修改部分程序，或把几个程序合并为一个程序。也可以把整数BASIC程序转换成CEC-BASIC等等。

关于使用EXEC命令来执行DOS命令和BASIC语句，DOS 3.3系统主盘上给出了一个BASIC程序："EXEC DEMO"作为范例，有兴趣的读者可以分析一下该程序所做的工作。

6.7.6 对二进制文件进行操作的DOS命令

DOS操作系统除了可以对BASIC程序文件、文本文件进行操作处理外，还能够对2进制文件进行操作。在文件目录中，2进制文件的类型标志是B文件。

2进制文件存放的是主机内存中的2进制信息副本，即把主机内存中某个地址段的信息以文件的形式保存在软盘上。这些文件信息可以是机器语言程序，也可以是2进制数据或者是图象信息等。2进制文件在软盘上的存储格式为：

地址　　长度　　2进制数据块

DOS操作系统对2进制文件提供了三条操作命令。即：BSAVE、BLOAD和BRUN。

1. 保存二进制文件（BSAVE）

用途：保存2进制文件。

格式：BSAVE f, Aa, Lj[,Ss][,Dd][,Vv]

说明：该命令将把内存中指定的一段2进制数据，以f为文件名，保存到指定的软盘上。

命令参数Aa指明了要存储的2进制数据在内存中的起始地址。a的值可以在0～65535之间。

命令参数Lj指明了该2进制数据块的字节长度。j的值可以在0～32767之间。

值得注意的是：在此命令中命令参数Aa和Lj均是必须的。参数a和j可以用10进制来表示，也可以用16进制来表示。用16进制表示时，在数值前面要冠以字符“$”。

例：用下面二个命令都可以创建一个名为“PICTURE的2进制文件，该文件将包含在主机内存中高分辨图象第二页内的图象信息。

```
BSAVE PICTURE, A $4000, L $2000
BSAVE PICTURE, A16384, L 8192
```

2. 装入二进制文件（BLOAD）

用途：装入2进制文件。

格式：BLOAD f [,Aa][,Ss][,Dd][,Vv]

说明：该命令的作用是将盘上的2进制文件f装入到主机内存的指定区域中。

执行该命令时，系统工作状态不会发生变化，也不会破坏内存中的BASIC程序，除非被装入的2进制文件侵入到BASIC程序所占用的存储空间。

命令参数Aa指明了2进制文件装入到内存区域的起始地址。a的值可以在0～65535之间(16进制则为$0～$FFFF)。当该命令参数省略时，其起始地址就是该文件由“BSAVE”

命令创建时所指明的起始地址。

例如，若要将PICTURE文件装回到原来的存储区中（即高分辨图象区的第二页）中，可用命令：

BLOAD PICTURE

BLOAD PICTURE, A $ 4000

BLOAD PICTURE, A16384

使用下列命令则将该图象信息装入到高分辨图象区的第一页。

BLOAD PICTURE, A $ 2000

BLOAD PICTURE, A8192

要注意，使用“BLOAD”命令将机器语言程序装入内存时，如果指定的存储区与该程序原来建立时的存区不一致时，该程序可能就不能运行了。

3. 装入并运行机器语言程序（BRUN）

用途：装入并运行机器语言程序。

格式：BRUN f[,Aa][,Ss][,Dd][,Vv]

说明：该命令首先将装入以f为文件名的2进制文件，然后转移（JMP）到该文件的第一个字节去执行程序。

在使用该命令时要做到以下几点才不致于引起系统的混乱和“挂起”。

（1）必须确保f是由机器语言程序所组成的2进制文件。

（2）该程序的入口地址是该文件中的第一个字节。

（3）如果在命令中使用了命令参数Aa，则参数a所给出的地址要和建立该文件时起始地址相一致。

6.7.7 辅助命令

1. 置监视方式（MON）

用途：置监视方式。

格式：MON[C][,I][,O]

说明：在用程序方式调用DOS命令时，DOS命令及主机与磁盘之间的信息交换均不在屏幕上显示出来，但在调试程序时，监视这些信息却是十分必须的。通过它我们可以找出程序中的问题。MON命令可以满足我们监视这些信息的要求。

在该命令中有三个参数可供我们选择，它们分别代表命令信息、输入信息和输出信息：

C显示所使用的DOS命令。

I 显示由磁盘读入主机的文本信息。

O显示由主机输出到文本文件中的信息。

这三个参数可以按任何顺序和任意组合形式出现，这取决于希望监视的信息。如果这三个参数均不出现，则MON命令不起任何作用。

例：下面这个例子将显示DOS命令和从主机写到软盘上的数据。

```
10  D $=CHR $ (4) : REM CTRL-D
20  DIM A $ (10)
30  NAME $="ANIMALS"
40  PRINT D$; "MON C, O"
50  PRINT D$; "OPEN"; NAME $
60  PRINT D$; "WRITE"; NAME $
70  FOR I=1 TO 3:READ A $: PRINT A $:
```

```
    NEXT I
80  PRINT D $; "CLOSE", NAME $
90  FOR I=1 TO 3000:NEXT I
100 PRINT D $; "OPEN", NAME $
110 PRINT D $; "READ", NAME $
120 FOR I=1 TO 3: INPUT A $ (I) : NEXT I
130 PRINT D $; "CLOSE", NAME $
140 DATA HORSE, CAT, DOG
150 END
```

其运行结果如下所示：

```
]RUN
OPEN ANIMALS
WRITE ANIMALS
HORSE
CAT
DOG
CLOSE ANIMALS
OPEN ANIMALS
READ ANIMALS
CLOSE ANIMALS
]
```

该命令执行后一直有效，除使用了RESET键或使用了NOMON、FP、INT等命令，或者在监控方式下执行了3DOG或3D3G等操作后，才会使得MON命令不起作用。

2. 解除监视方式（NOMON）

用途：解除监视方式。

格式：NOMON[C][,I][,O]

说明：该命令将撤消对其参数所指明的DOS操作信息的监视。参数的含义和使用方法和MON命令完全相同。

3．置输出设备选择（PR#）

用途：置输出设备选择。

格式：PR#S

说明：该命令用来选择输出设备，其中参数S表示该设备所占用的槽口号，S值可以是0～7之间的整数。

执行该命令后，系统的输出信息都将送到S值所指明的槽口设备上处理。对于中华学习机来说，S＝0表示输出为屏幕显示，S＝3表示进入汉字系统，S＝6表示从软盘上引导DOS操作系统。

例如，如果我们在中华学习机扩充槽内接有打印机，且槽口号定义为1号槽。我们要打印数据，可按下列程序执行操作。

```
10  PRINT CHR $ (4) ; "PR#1"
20  A$="ABCDEFG0123456789"
30  FOR I=1 TO 3:PRINT A $:NEXT I
40  PRINT CHR $ (4) ; "PR#0"
50  END
```

4．置输入设备选择（IN#）

用途：置输入设备选择。

格式：IN#S

说明：该命令用来选择输入设备，其中参数S表示该设备所占用的槽口号，S值可以是0～7之间的整数。

执行该命令后，系统的输入信息都将由S值所指定的槽口设备所提供，对于中华学习机来说，S＝0表示输入设备为键盘，S＝3表示进入汉字系统，S＝6表示从软盘上引导DOS操作系统。

5．置文件缓冲区的大小（MAXFILES）

用途：置文件缓冲区的大小。

格式：MAXFILESn

说明：该命令用来设置文件缓冲区的个数，参数值n可以是1～16之间的整数。

一个文件缓冲区占用595个字节的内存空间，每一个被使用的文件均占用一个文件缓冲区，对某个文件执行读操作时，首先从盘上将一个扇区的信息（256个字节）读入到该文件的缓冲区中，然后再将其中的一部分送给程序。如果对某个文件执行写操作时，也是把信息送到文件缓冲区，待写满256个字节时再送到磁盘上。

文件缓冲区的个数决定了程序一次可以打开的文件个数。启动DOS时，系统将自动执行"MAXFILES 3"命令，把文件缓冲区的数目定为3个。所以它一次最多可以打开的文件个数只能是3个文件。如果一次打开的文件个数大于MAXFILES命令中指定参数值n时，将给出信息：NO BUFFERS AVAILABLE。

例如，我们在主机内存中打入如下程序。

```
5    DIM A $ (10) , B$ (10)
10   D$ = CHR $ (4) :N1 $ = "ABC": N2$
     = "XYZ"
20   PRINT D S; "OPEN"N1 $
25   PRINT D $; "DELETE", N1 $
30   PRINT D $; "OPEN"; N1 $
35   PRINT D $; "WRITE", N1 $
40   FOR I = 1 TO 5: READ A $ (I) :PRINT A $ (I) ,
     NEXT I
50   PRINT D $, "OPEN", N2 $
```

```
55   PRINT D $; "DELETE"; N2 $
60   PRINT D $; "OPEN"; N2 $
65   PRINT D $; "WRITE"; N2 $
70   FOR I=1 TO 5:B $=LEFT $ (A$ (I) , 1) :
       PRINT B $ : NEXT I
80   PRINT D $; "CLOSE"
90   DATA ABC, BCD, CDE, DEF, EFG, GHI
100  END
```

在打入RUN命令前，我们执行MAXFILES1命令。在DOS监视方式下，我们会得到如下信息：

```
]MAXFILES1
]RUN
OPENABC
DELETEABC
OPENABC
WRITEABC
ABC
BCD
DEF
EFG
GHI
OPENXYZ

NO BUFFERS AVAILABLE

Break in 50
```

遇到NO BUFFERSAVAILABLE错误时，其解决的唯一办法是关闭文件，我们把上面这段程序稍加改造，其程序及运行结果如下：

```
5    DIM A$ (10) , B$ (10) ,
10   D $=CHR $ (4) :N1 $="ABC":N2 $
```

```
      ="XYZ"
15   ONERR GOTO 110
20   PRINT D $; "OPEN"; N1 $
25   PRINT D $; "DELETE", N1 $
30   PRINT D $; "OPEN"; N1 $
35   PRITN D $; "WRITE"; N1 $
40   FOR I=1 TO 5: READ A $ (I)
     PRINT A $ (I) :NEXT I
50   PRINT D $; "OPEN"; N2 $
55   PRINT D $; "DELETE"; N2 $
60   PRINT D $; "OPEN"; N2 $
65   PRINT D $; "WRITE"; N2 $
70   FOR I=1 TO 5:B $=LEFT $ (A $ (I) , 1)
     PRINT B $:NEXT I
80   PRINT D $; "CLOSE"
90   DATA ABC, BCD, CDE, DEF, EFG, GHI
100  END
110  PRINT D $; "CLOSE":RESUME

] MAXFILES1

] RUN
OPENABC
DELETEABC
OPENABC
WRITEABC
ABC
BCD
CDE
DEF
EFG
OPENXYZ CLOSE
```

```
OPENXYZ
DELETEXYZ
OPENXYZ
WRITEXYZ
A
B
C
D
E
CLOSE
```

注意：在执行 MAXFILES 命令时，系统将修改 HIMEM：的值，所以放在 HIMEM 之下的整数 BASIC 程序和 CEC-BASIC 程序的字符串变量有可能被破坏。故在使用 MAXFILES 命令时建议：对于 CEC-BASIC 程序中使用 MAXFILES 命令时，应把它作为程序的第一条语句，放在任何串变量赋值语句和字串数组定义之前。对整数 BASIC 程序，应放在装入 BASIC 程序之前，使用该命令。

6. BASIC 程序的链接（CHAIN）

（1）整数 BASIC 程序的链接

格式：CHAIN f [, Ss] [, Dd] [, Vv]

说明：该命令只适用于整数 BASIC 程序之间的链接，链接操作的目的是装入运行第二个 BASIC 程序时，第一个程序运行的变量结果仍保留在内存中，可以为第二个程序所使用。

例如，我们可以建立如下二个程序，分别取名为程序 A 和程序 B，保存到工作盘上。

程序A：

```
5  DIM A (10)
10  FOR I=1 TO 5:A (I) =I*2: PRINT A (I) , :
      NEXT I:PRINT
20  PRINT "CHAIN B"
30  END
```

程序 B:

```
10  FOR J=1 TO 5: A (J) =A (J) +J: PRINT
      A (J) , : NEXT J: PRINT
20  END
```

打入命令 MON C 和 RUN ，可得到如下结果：

```
>MON C
>RUN A
2      4      6      8      10
CHAINB
3      6      9      12     15
```

(2) CEC-BASIC 程序的链接

在 DOS 3.3 操作系统中没有提供 CEC-BASIC 程序之间的链接操作，但在 DOS 3.3 系统主盘上提供了 CHAIN 程序，利用该程序，可以完成 CEC-BASIC 程序之间的链接。

使用时首先将 CHAIN 程序复制到我们的工作盘上，然后在第一个程序的末尾加上以下两条语句即可：

```
PRINT CHR $ (4) ; "BLOAD CHAIN, A520"
CALL 520"第二个程序名"
```

例如，我们将上例中的两个程序按新要求改正如下，并分别取名为AA和AB，保存到工作盘上。

程序AA：

```
5  DIM A (10)
10  FOR I =1 TO 5:  A (I) = I*2: PRINT A (I) , :
    NEXT I:PRINT
20  PRINT CHR $(4) ; "BLOAD CHAIN, A520"
30  CALL 520 "AB"
```

程序AB：

```
10  FOR J= 1 TO 5: A (J) =A (J) +J:
    PRINT A (J) , : NEXT J: PRINT
20 END
```

打入RUN AA命令后，屏幕将显示：

```
] RUN
2          4          6
8          10


3          6          9
12         15
```

6.8 DOS3.3 操作系统的实用程序

6.8.1 DOS 3.3 系统主盘

DOS 3.3 系统主盘是一张很特别的系统引导盘，它可以在主存为 16～48KB 的 Apple Ⅱ机上运行，也可以在主存为 64KB 的中华学习机、Apple Ⅱe等机器上运行。它除了含有 DOS 3.3 磁盘操作系统外，在它的软盘上还含有许多实用程序。使用这些程序，可以能够运行整数 BASIC 程序，进行软盘片的复制，13扇区软盘的引导，主系统盘的建立等等。此外，盘上还有不少与DOS 及 BASIC 程序的使

用有关的示范程序等。

下面给出该软盘的目录清单及简单的注释：

DISK VOLUME 254

*	A	008 HELLO	系统启动后执行的第一个程序
*	I	018 ANIMALS	猜字游戏程序
	T	003 APPLE PROMS	由 RANDOM 程序建立的随机文件
*	I	006 APPLESOFT	装入 APPLESOFT 解释程序的实用程序
*	I	026 APPLEVISION	动画图形表演程序
*	I	017 BIORHYTHM	计算人体生物曲线的游戏程序
*	B	010 BOOT13	13扇区软盘的引导程序
*	A	006 BRIAN'S THEME	高分辨率作图示范程序
*	B	003 CHAIN	APPLESOFT链接程序
*	I	009 COLOR DEMO	彩色示范程序
*	A	009 COLOR DEMOSOFT	彩色示范程序
*	I	009 COPY	软盘复制程序
*	B	003 COPY OBJ0	软盘复制程序所要用到的RWTS
*	A	009 COPYA	软盘复制程序
*	A	01 EXEC DEMO	EXEC 命令的示范程序
*	B	020 FID	软盘文件管理程序
*	B	050 FPBASIC	APPLESOFT 解释程序，由 APPLESOFT 程序装入
*	B	050 INTBASIC	整数 BASIC 解释程序由 HELLO 程序装入
*	A	003 MAKE TEXT	建立文本文件的示范程序
*	B	009 MASTER CREATE	建立主系统盘的实用程序
*	B	027 MUFFIN	将13扇区的文件转换成16扇区文件的实用程序
*	A	051 PHONE LIST	存储电话号码的应用程序

	A	010 RANDOM	使用随机文件的库存管理示范程序
*	A	013 RENUMBER	对 APPLESOFT 程序可重新编行号的实用程序
*	A	039 RENUMBER INSTRUCTIONS	RENUMBER 实用程序的使用介绍
*	A	003 RETRIEVE TEXT	取文本文件的实用程序

6.8.2 HELLO

HELLO 程序是用 DOS 3.3 系统主盘启动后装入运行的第一个 BASIC 程序。该程序将检查主机的内存，如果是已扩充到 64KB RAM 主机系统，则通过装入 INTBASIC 文件，把整数 BASIC 解释程序和小汇编程序装入内存。这样系统便可以使用 INT 命令来进入整数 BASIC 状态。

在运行 HELLO 程序后，屏幕上将出现下列信息：

```
DOS VERSION 3.3                08/25/80
APPLE II PLUS OR ROMCARD SYSTEM
MASTER
(LOADING INTEGER INTO LANGUAGE
 CARD)
```

通过第二行信息，我们知道主机的内存已扩充为64KB而第三行信息则告诉我们整数 BASIC 解释程序已装入内存。

对于中华学习机或 CEC-I A 机器来说，主机内存固化的是 CEC-BASIC 解释程序，要使用整数 BASIC 语言，则必须要装入整数 BASIC 程序，启动 DOS 3.3 系统主盘，或运行 DOS 3.3系统主盘上的 HELLO 程序，是装入整数 BASIC 解释程序的最简单方法之一。

6.8.3 APPLESOFT

该程序是为早期固化了整数BASIC解释程序的APPLE Ⅱ机用户设计的，其作用是通过将 FPBASIC 2进制文件装入扩充为 64 KB 的机器中，使得该机器可以运行APPLESOFT程序。对于中华学习机，由于都固化了CEC-BASIC解释程序，所以该程序一般不被使用。

6.8.4 COPYA和COPY

COPYA和COPY程序是软盘片复制程序，它们分别是用APPLESOFT语言和整数BASIC语言编写的，其功能和操作方法完全相同。它们都需要通过装入COPY.OBJ0 2进制文件来完成软盘片的复制工作。

在盘片的复制过程中，被复制的盘片称为“源盘”，而复制后的盘片称为“复制盘”。复制的最终目标是把“源盘”上的全部信息，一道一道地抄写到“复制盘”上，这样得到的复制盘是与源盘毫无区别的副本。为了防止DOS 3.3系统主盘在使用中被破坏，我们建议用户在使用前，一定要复制一个副本，保存起来。

软盘片的复制过程如下：

（1）将DOS3.3系统主盘插入软盘驱动器，打入命令：RUNCOPYA（或RUNCOPY）

（2）屏幕上将给出信息：

```
APPLE DISKET TE DUPLICATION PROGRAM
ORIGINAL     SLOT:   DEFAULT=6
```

此时，依次回答源盘所在驱动器的槽口号和设备号分别为6和1。

（3）屏幕上将紧接着询问复制盘所在的驱动器的槽口号和设备号。

DUPLICATION SLOT: DEFAULT=6

如果我们是在单盘驱动系统下工作的，回答仍依次为6和1。

如果我们是在多盘驱动系统下工作的，则回答复制盘所在驱动器的槽口号和设备号。

（4）全部应答完毕后，屏幕将呈现如下状态：

```
APPLE DISKETTE DUPLICATION PROGRAM
ORIGINAL  SLOT:  6
          DRIVE: 1
DUPLICATESLOT:   6
          DRIVE: 1
——PRESS'RETURN'KEY TO BEGIN COPY——
```

（5）按 RETURN 键，使开始进行盘片的复制。

如果是在单盘驱系统下，程序将告诉我们何时插入源盘，何时插入复制盘。

如果是在多盘驱系统下，程序也会通过屏幕显示告之我们程序执行到哪一步了，在做什么操作。

（6）盘片复制完毕后，主机将显示信息：

DO YOU WISH TO MAKE ANOTHER COPY?

如果需要则打 Y 键，否则按 N 键退出该程序执行。

这里值得注意的是：

（1）盘片复制过程中，如果没有适时地插入源盘，或者盘驱动器门没关上，或者源盘片是已被加密的，或者源盘

中已有文件被损坏，那么可能会出现信息：

* * * * * UNABLE TOREAD * * * * *

（2）如果没有适时地插入复制盘，或者复制盘上贴有写保护，或者错误地指定了复制盘所在驱动器占有的槽号和设备号，将出现如下信息：

* * * * * UNABLE TO WRITE * * * * *

（3）为了防止操作上的失误而导致源盘片被损坏，建议用户在执行复盘操作前，对源盘片贴上写保护。

6.8.5 FID

FID程序是用于磁盘文件管理的实用程序，它通过菜单选择对指定的磁盘文件进行有关操作。

当程序要求我们提供一个文件时，我们可以输入一个确定的文件名。也可以使用符号“=”作为“百搭”字符，字符“=”可以代替文件名中的任何字符串。例如：使用文件名“AB=CD”,程序将对所有以“AB”开头以“CD”结尾的文件名代表的文件执行操作。使用文件名“=N=”，则选择所有文件名中包含字符“N”的文件名执行操作，使用文件名“=”则选择对所有文件执行操作。

如果回答文件名时使用了百搭字符，程序会提问：DO YOU WANT PROMPTING?

（1）如果回答Y，则表示程序在选择到与给出的文件名相匹配的文件名时，将显示该文件名，并等待我们去证实，如果对该文件执行操作，打入Y键，否则打入N键。如果中止后继续操作则打入Q键。

（2）如果回答N，则表示程序将选择对所有与给出的文件名相匹配的文件执行操作。

该程序的执行方法：

打入命令：BRUN FID，屏幕上将显示选择菜单：

```
* * * * * * * * * * * * * * * * * * * *
*          APPLE ][  FILE DEVELOPER           *
*               FID VERSION M                 *
*                                             *
*      COPYRIGHT 1979 APPLE COMPUTER INC      *
* * * * * * * * * * * * * * * * * * * *
CHOOSE ONE OF THE FOLLOWING OPTIONS

    (1)  COPY FILES
    (2)  CATALOG
    (3)  SPACE ON DISK
    (4)  UNLOCK FILES
    (5)  LOCK FILES
    (6)  DELETE FILES
    (7)  RESET SLOT & DRIVE
    (8)  VERIFY FILES
    (9)  QUIT

WHICH WOULD YOU LIKE? 9
```

上述菜单中1～9为各种不同的操作，可以通过键入相应的数字，并按回车键来选择所需的操作。

(1) COPY FILES

将源盘上的指定文件复制到复制盘上，它与COPYA程序所不同的是：它仅对指定文件进行复制，而不是对整张软盘进行复制。复制盘必须是经过初始化操作的盘。

(2) CATALOG

将执行CATALOG列文件目录的操作

(3) SPACE ON DISK

用于检查指定盘片的自由空间

(4) UNLOCK,LOCK,DELETE,VERIFY

执行这类命令，将对所指定文件执行相应的操作。

(5) RESET SLOT & DRIVE

在执行前面几项操作时,系统仅询问一次与操作有关盘片所在驱动器的槽口和设备号，然后系统将默认这些参数，若用户想改变驱动器的槽号和设备号参数时，可以使用该参数。此时，系统将给出下列信息：

RESET SLOT & DRIVE

DONE

PRESS ANY KEY TO CONTINUE

当再次选择有关操作命令时，系统将象第一次使用该操作一样询问有关操作的参数。

(6) QUIT

该项选择将退出 FID 程序的执行。

6.8.6 MUFFIN

该程序的作用是将13扇区软盘片上的文件通过程序转换写到16扇区的软盘片上。

对于在DOS 3 .2版以前的操作系统上开发的软件，均是以13扇区数据格式写到软盘上的，它是不能在DOS 3.3操作系统下运行的，否则将给出信息:UNABLE TO READ /WRITE。要想使得它能在中华学习机上使用，必须把这些文件转换成16扇区格式，存放到经过DOS3.3初始化操作后的软盘上，MUFFIN程序将完成此项操作。

该程序的操作步骤如下：

(1)插入 DOS 3.3 系统主盘，打入命令：BRUN MUFFIN。屏幕上将出现信息：

```
* * * * * * * * * * * * * * * * * * * *
*     APPLE][DOS 3.2 TO 3.3 CONVERTER     *
*                                         *
*          MUFFIN VERSION D               *
*                                         *
*   COPYRIGHT 1979 APPLE COMPUTER INC     *
* * * * * * * * * * * * * * * * * * * *
CHOOSE ONE OF THE FOLLOWING OPTIONS
        (1) CONVERT FILES
        (2) QUIT
     WHICH WOULD YOU LIKE? 2
```

(2) 打入数字1，进行文件的转换。

(3) 回答源盘和转换后的复制盘所在驱动器的槽号和设备号。

(4) 回答系统询问所要转换的文件名，可以输入确定的文件名，也可以用百搭字符（=）让系统帮助查找相匹配的文件进行转换。

(5) 取出 DOS3.3系统主盘，插入13扇区的源盘，以及16扇区的复制盘，通过屏幕的提示，便可完成文件的转换。

转换文件的各项操作步骤与 FID 程序的 COPY 操作基本一样。所要注意的是：复制盘一定是经过DOS3.3初始化操作后的软盘。

6.8.7 BOOT 13

该程序是13扇区软盘的引导程序，如果我们拿到的是一张在 DOS 3.2操作系统上开发的软盘，而想要直接运行该

盘上的软盘，可用 BOOT 13程序进行引导。

该程序的操作方法如下：

插入 DOS 3.3 系统主盘，打入命令：BRUN BOOT13，屏幕上将出现信息：

13-SECTOR BOOT UTILITY

SLOT TO BOOT FROM (DEFAULT=6)?

此时，取出 DOS 3.3系统主盘，插入13扇区软盘，并打入 RETURN 键，即可引导13扇区软盘进行工作。

6.8.8 MASTER CREATE

在DOS操作系统上用 INIT命令初始化新盘片时，得到的是一张从属DOS系统盘，所谓从属系统盘是指操作系统装入地址与主机内存的大小有关。如果你是在内存为48KB的机器上初始化后得到的盘片，不能在内存为16KB或32KB的APPLE Ⅱ机上引导操作系统。而对于DOS 3.3系统主盘来说，它可以在内存为16KB～64KB的所有 APPLE Ⅱ机上启动运行，我们说 DOS 3.3系统主盘是一张主导盘（或称主盘）。

主盘和从属盘从外表上是看不出来的，甚至在机器运行过程中也看不出来，但利用 MASTER CREATE 程序修改问候程序，以及通过软盘上注明的标签才可知道。

MASTER CREATE 程序具有二项功能：

（1）把一个从属盘（它的DOS 与内存大小有关）转换成主导盘（它的 DOS 是自定位的，可以在任何大小的系统上有效地利用内存）。

（2）可以重新指定 DOS 操作系统引导后所装入运行

的程序名称。

MASTER CREATE 程序的使用步骤：

（1）插入 DOS 3.3系统主盘，并打入命令：BRUN MASTER CREATE.屏幕上将出现如下信息：

DOS 3.3 MASTER-CREATE UTILITY
COPYRIGHT 1980 BY APPLE COMPUTER INC
ALL RIGHTS RESERVED
〈NOW LOADING DOS IMAGE〉

（2）如果插入的盘片不是主导盘或不是DOS 3.3系统盘，屏幕将出现以下二种信息之一，换上DOS 3.3系统主盘，并按 RETURN键重新引导 DOS。

情况1：

IMAGE OF DOS 3.3〈MASTER〉IS NOT
AVAILABLE.CHECK INSTRUCTIONS.

INSERT A SYSTEM DISKETTE AND PRESS
〔RETURN〕TO REBOOT DOS

情况2：

UNABLE TO READ IMAGE.
INSERT A SYSTEM DISKETTE AND PRESS
〔RETURN〕TO REBOOT DOS

至此，执行MASTER CREATE程序失败，需要重新运行。

（3）当屏幕上出现如下提示时，请你回答该盘片DOS操作系统引导后，装入运行的第一个BASIC程序的程序名，你可以输入执行INIT命令操作时的文件名，也可以是其它文件名。这样用户可以根据需要指定任意的程序作为"问候"程序。

PLEASE INPUT THE"GREETING"PROGRAM'S

FILE NAME：

（4）此时，屏幕将出现如下信息：

REMEMBER THAT"MASTER"DOES NOT
CREATE
THE"GREETING"PROGRAM, OR PLACE IT IN
THE DISK DIRECTORY
THIS IS THE FILE NAME THAT WILL BE
PLACED WITHIN THE IMAGE：
HELLO
PLACE THE DISKETTE TO BE MASTERED IN
THE DISK DRIVE.
PRESS [RETURN] WHEN READY
NOTE： IF YOU WANT A DIFFERENT FILE NAME,
PRESS [ESC]。

从软盘驱动器中取出DOS3.3系统主盘，插入要转换的从属盘，并按RETURN键。

（5）转换完毕后，程序将通过如下的提示，让你选择是继续转换其它从属盘，还是退出该程序的执行，

THE DISKETTE HAS BEEN UPDATED,
YOU MAY
REMOVE IT AT THIS TIME.
IF YOU WISH O"MASTER" ANOTHTER
DISKETTE, PRESS [RETURN] .
OTHERWISE PRESS [ESC] TO EXIT "MASTER"

注意：当你执行第三步操作时，回答的文件名并不放在该盘片的目录中，它只是告诉软盘的DOS,每次引导DOS后就运行该程序。如果回答的程序文件不在该软盘上，每次用该盘片启动后，将得到信息：FILE NOT FOUND。

第七章 打印机的安装与使用

CEC-ⅠA型中华学习机配有并行打印机接口及打印机驱动软件。通过该接口可以与多种型号的打印机连接使用，实现 ASCII 字符的输出，高分辨图形的硬拷贝和汉字的打印等功能。本章将以MX-80Ⅲ型打印机为例，介绍九针打印机的连接安装及BASIC程序下的打印机使用方法。

7.1 打印机的选择

CEC-ⅠA型中华学习机可以配接具有Centronics接口的多种型号的打印机，可以是九针的，十六针的，二十四针的，甚至可以是激光打印机等等。从系统设计的角度考虑，中华学习机是面向中小学校、家庭及个人使用的低档微机系统，它以价格低廉、功能齐全、使用方便为目标。因此，在固化软件的设计上，特别是高分辨图形的硬拷贝和汉字打印程序等等，是针对日本EPSON公司生产的九针打印机的使用而设计的，凡是与MX-80Ⅱ型九针打印机相兼容的九针打印机均可以与中华学习机相连接使用。

推荐用户选用的九针打印机的型号有：

(1) MX系列：MX-80Ⅱ、MX-80Ⅲ、MX-100Ⅱ、MX-100Ⅲ

(2) FX系列：FX-80、FX-100

表 7-1 几种常用打印机基本特性表

	打印机型号	MX-80 II	MX-80 III	MP-80 III	RX-80	LX-800	FX-100
基本特性	打印行宽	80/40/132★	80/40/132★	80/40/132★	80/40/137★	80/40/132★	180/68/233★
	打印速度	80 CPS	80 CPS	80 CPS	100 CPS	180 CPS☆	100CPS
	换行时间	200ms/1	200ms/1	200ms/1	200ms/1	95ms/1	200ms/1
	控制命令	23条	41条	42条	53条	75条	53条
	打印字符集	96个▲	96个●	94+128个	96+128个◆	96+128个■	96+128个◆
	行间距控制	3种	5种	5种	5种	5种	5种
	走纸方式	链 式	链 式	链 式	链 式	链式、滚轮	链式、滚轮
	打印方向	双向字符	双向字符	双向字符	双向字符	双向字符	双向字符
	电源功率	100W	100W	100W	70W	70W	100W
工作环境	环境温度	5°C～35°C					
	相对湿度	10%～80%					
备注	★ 正常/放大/压缩打印字体。 ▲ ASCII字符集+四国文字。 ◆ ASCII字符集+十一国文字。 ☆ 机内含有3K字节缓存RAM。 ● ASCII字符集+八国文字。 ■ ASCII字符集+十二国文字。						

（3）LX系列：LX-800、LX-1000

（4）其它型号：MP-80Ⅲ、RX-80Ⅲ

（5）兼容机型号：CP-80⁺、YAMATO，Super

这类打印机的共同特点是：机器的连接安装方法大致相同，工作环境相同，基本的控制命令码相兼容，使用方法也大致相同。在下面的介绍中，仅以MX-80Ⅲ型打印机为例，说明如何操作和使用打印机，对于其它型号打印机的使用方法，不同之处可参阅随机所带的操作使用手册。

为了方便用户对各类打印机的基本特性有所了解和选择使用，可参阅表7-1所列的基本特性表。

7.2 打印机的安装与检查

7.2.1 打印机的安装方法

用户启封打印机包装箱后，可以见到如图7-1所示的几件东西（见图7-1）。

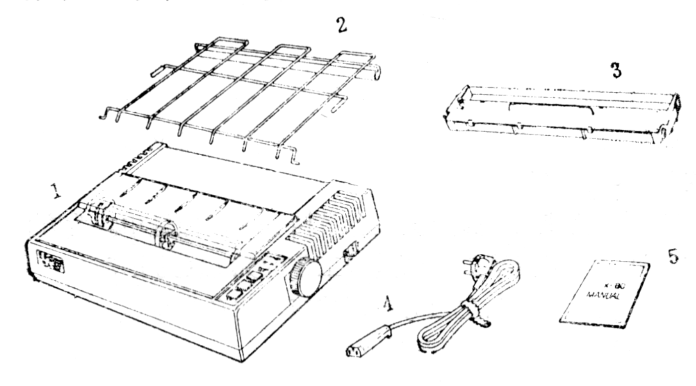


图 7-1

其中包括：

（1）MX-80Ⅲ型点阵式九针打印机

（2）打印纸分隔架

（3）色带盒

（4）电源线

（5）打印机操作手册

打印机的连接安装过程一般按下列步骤进行：

1。安装色带盒

（1）打开打印机的上盖板，并取下放置在一边（见图7-2）。

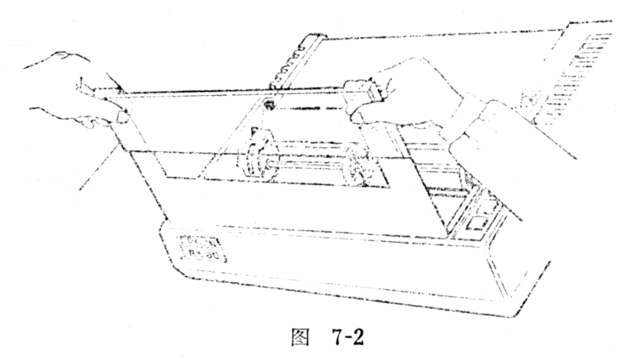


图 7-2

（2）用手捏住色带盒，并沿着色带盒的定位槽口将色带盒放入打印机（见图7-3）。

（3）将色带放在打印头和色带档板的间隙缝中。为方便安装起见，可用铅笔尖协助这一工作的完成(见图7-4)。

（4）旋紧色带盒左边的馈带旋钮，使色带拉紧。

2。安装打印机连接电缆及电源线

（1）将打印机连接电缆线上的20线插头插入主机的插座

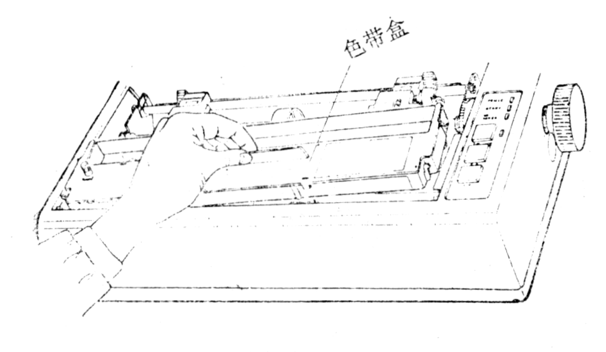


图 7-3

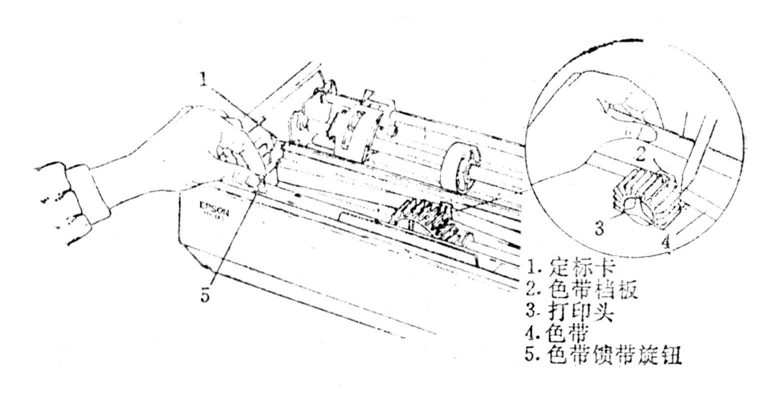


图 7-4

上。注意：矩形插头的凸出部分和插座的缺口方向要一致（见图7-5）。

（2）将电缆线的36线插头插入到打印机的插座上，并扣上锁定装置（见图7-6）。

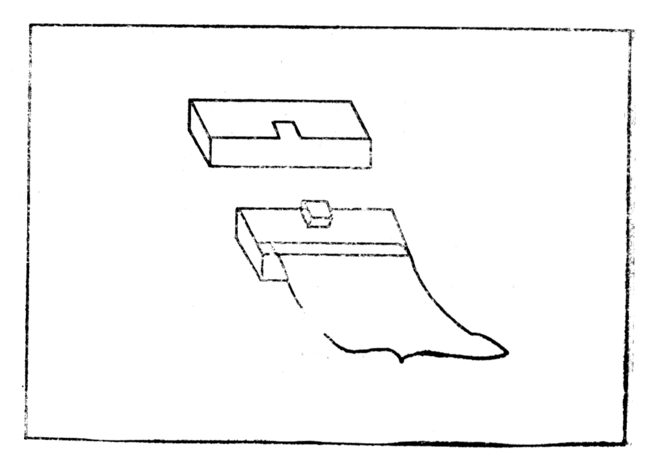


图 7-5

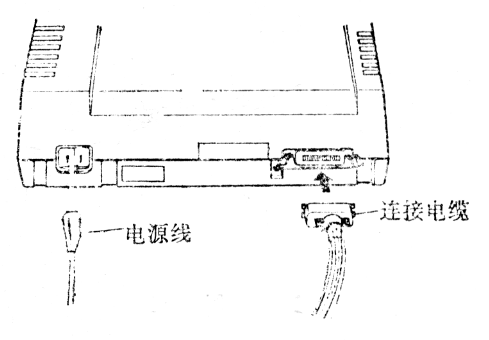


图 7-6

（3） 将电源线插头插入打印机上。注意：在插电源线以前，应置打印机的电源开关处于关闭状态。

3．安装打印纸

（1） 将打印纸分隔架安装在定位孔内（图7-7）。

（2）将打印纸从分隔架下的滚筒上推入。注意：一般先推开定标卡和打开链轮走纸齿轮的上盖，打印纸推进的路径

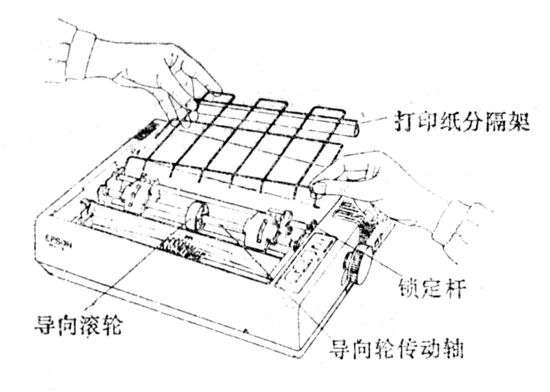
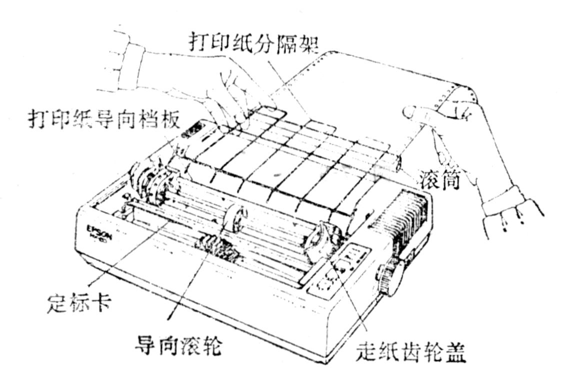


图 7-7



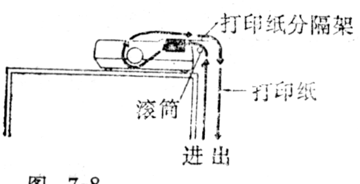


图 7-8

应按图7-8所示的箭头流向进行。

（3）向后推开链轮走纸齿轮的锁定控制杆，使链轮可以左右移动（图7-9）。

（4）将打印纸两边的孔对应放在链轮的走纸齿上，并将盖板放下，若打印纸能平整地压在链轮上，即可推上锁定控制杆。否则可调整一下链轮位置使打印纸平整后再推上锁定控制杆（图7-10）。

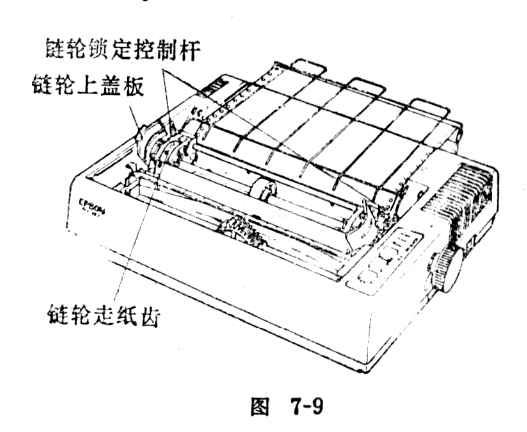


图 7-9

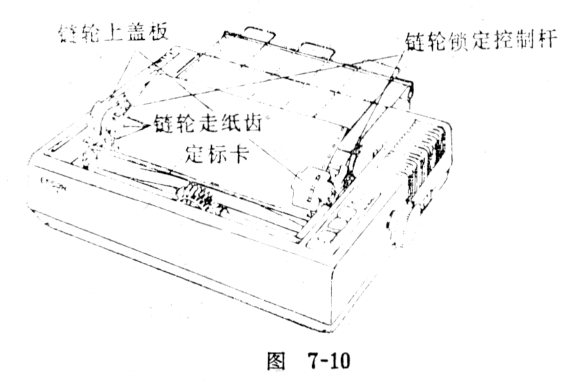


图 7-10

(5) 将定标卡推上，压住打印纸，并盖上打印机的盖板（图7-11）。

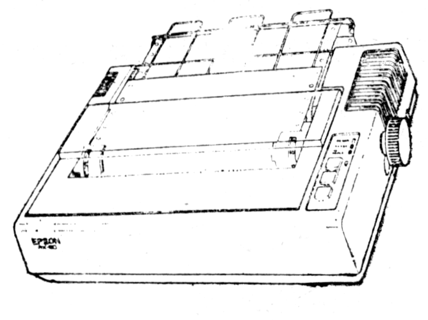


图 7-11

注意事项

(1) 打印机在工作台的安装位置及打印纸的放置应按图7-12进行。

(2) 打印机上有一调整打印头间隙的控制杆以适应不同打印份数的需要（图7-13）。

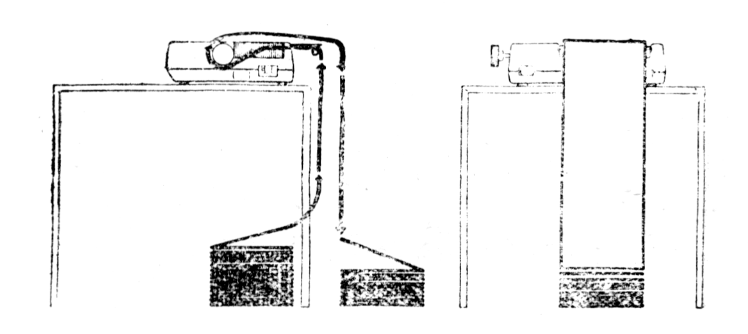
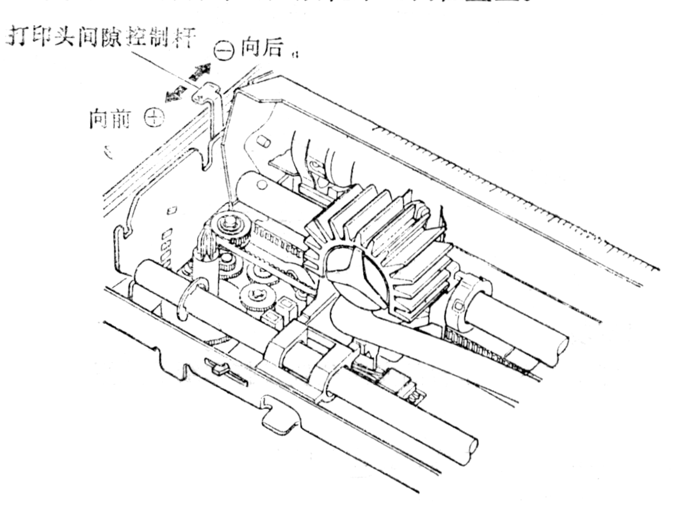


图 7-12

当打印头间隙控制杆向打印机后面推时，打印头与后档板之间的间隙减小，当打印头间隙控制杆向前拨时，间隙增大。一般使用单层纸进行打印时应放在第四档位置上，当夹上复写纸进行打印时应放在第七档位置上。



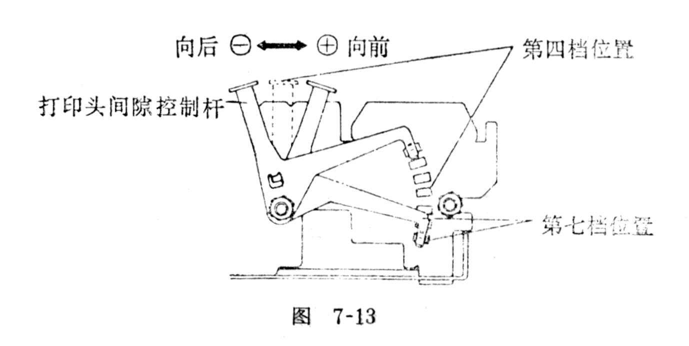


图 7-13

7.2.2 打印机的检查

通过下面的检查，可以得知打印机的安装是否正确，打印机本身能否正常工作。

1. 自检功能的检查

自检功能是打印机本身提供的一种最基本的测试手段。通过自检功能的检查可以得知：色带盒的安装、打印纸的安装是否正确，打印头的操作是否正常，打印质量如何，打印机的传动部分工作是否正常等等。

自检操作方法是：在打印机各部分安装正确，并装有打印纸的前提下，按住打印机控制面板上的LF开关，再打开打印机的电源开关，即启动打印机自检程序的执行，打印机内部所提供的ASC II字符将打印在打印纸上。关闭打印机电源开关后，自检操作立即停止。

如果在自检过程中打印出来的ASC II字符没有残字破字，打印行间隔一致，打印头没有抖动现象，机器不发生异常声音，则表示自检正确。

如果在按下LF开关，再打开电源开关后不能打印出任何信息，则表示没有能启动自检程序的执行，此时需要检查一下，打印纸安装是否正确，电源线是否接好，插头是否松动等等。

如果在自检过程中，打印的字符残破不全，或走纸不匀，或打印了一些胡乱的信息等等，则说明打印机本身发生了故障，需要进行修理。

2. 联机检查

联机检查是在确认主机和打印机本身工作都正常的情况

下进行的，通过它可以得知打印机与主机之间的连接电缆线安装是否正确，主机的打印机接口电路工作是否正常等等。

联机检查需要在打印机安装正确，主机和打印机之间已安装了连接电缆线之后才进行的。最简单的联机检查是在主机系统已经启动，并处于BASIC提示状态下进行的：

(1) 打开打印机的电源开关，此时除PAPER OUT指示灯不亮之外，所有的指示灯都亮。

(2) 在BASIC状态下，键入PR#1〈CR〉命令，将使得打印机连通，屏幕上将出现提示符。

(3) 键入命令PRINT "1234567890ABCD"〈CR〉后，屏幕和打印机将出现以下信息：

```
]PRINT"1234567890ABCD"
1234567890ABCD
```

(4) 键入命令PR#0〈CR〉后，将断开打印机。

在执行上述操作过程中有几点需要注意：

(1) 如果执行到第(2)步时，主机不出现任何提示信息，则说明系统发生了"挂起"故障。此时可检查打印机的READY指示灯是否为亮，如果不亮，可按打印机控制面板上的ON LINE开关，如果仍不能奏效则说明打印机接口电路发生了故障，或者打印机的连接电缆没有安装正确，需要仔细检查后重新安装。

(2) 如果在执行过程中，打印的字符与指定的字符不一致，或打印的字符混乱，则说明打印机的连接电缆线有可能发生了断线、碰线的故障。

(3) 如果在执行过程中，打印的字符发生漏字少字现象，则说明打印机驱动接口电路有可能发生了故障等等。

(4) 在发生系统“挂起”等故障时，可先用CTRL-RESET键强行将打印机断开，再对各部分进行检查，必要时需关闭打印机开关和主机开关，对各部重新进行安装和检查。

7.3 打印机的基本操作与使用

7.3.1 打印机的控制开关与指示灯

在打印机的控制面板上装有三个控制开关和四个状态指示灯，而打印机的右侧装有电源控制开关（图7-14）。

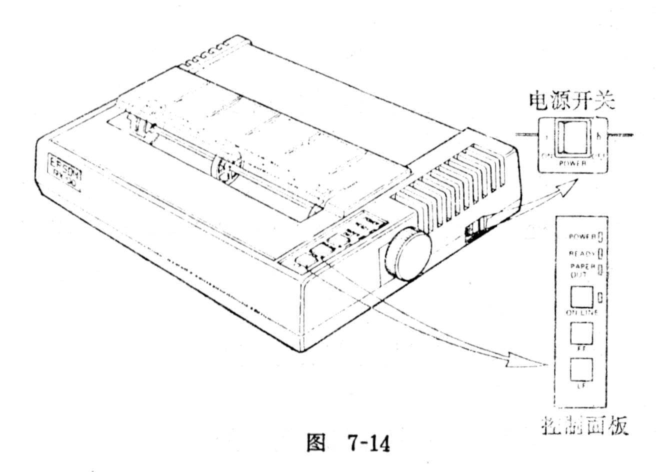


图 7-14

1. 控制开关

(1) 电源开关（POWER）：使得打印机开始工作的控制开关。

进行打印机安装或对打印机故障进行检查时，要注意使

得电源开关处于OFF状态，只有当需要使用打印机进行输出时才使得电源开关处于ON状态。在打印机控制面板上，有一个标有POWER字样的指示灯与之相对应，当该指示灯为亮时表示打印机已接通了电源。

（2）ON LINE开关：联机/脱机控制开关。

当此开关右边的状态指示灯为亮时表示打印机处于联机状态，打印机可受控于与之相连的中华学习机主机进行输出打印等等。按一下该开关，指示灯熄灭，它表示打印机处于脱机状态，打印机将不接受主机送来的任何命令，也不会进行打印输出，而只能接受打印机控制面板上的FF开关或LF开关所发送的操作指示。再按一下该开关，打印机又恢复成联机状态。

（3）FF开关：换页走纸控制开关。

当打印机处于脱机状态时，按一下此开关，将使得打印纸上卷到下一页的第一行上。

（4）LF开关：换行走纸控制开关。

当打印机处于脱机状态时，按一下此开关，将使得打印纸上卷一行。

2．状态指示灯

（1）POWER：表示打印机已接通了交流电源，处于工作状态。

（2）READY：表示打印机空闲，处于可接受数据和命令状态。

（3）PAPER OUT：表示打印纸用完或没有安装好。当打印机出现故障；例如打印色带断了或卡住了，打印纸卡纸等也会出现该指示。

(4) ON LINE：表示打印机处于可联机操作状态。

7.3.2 打印机的连通与断开

CEC-ⅠA中华学习机和打印机是二个独立的设备，当两者都处于加电状态时，学习机上的输出信息并不能从打印机上进行输出，它需要通过联机和断开命令才使得主机对打印机发生控制。

在不同的系统状态下的联机命令是不同的，对于汉字系统下的打印机接通和断开方法详见“汉字系统的使用”。下面介绍的是在BASIC状态和监控状态下的打印机接通和断开的方法。

BASIC状态下，接通：PR#1 〈CR〉

断开：PR#0 〈CR〉

监控状态下，接通：1 CTRL-P 〈CR〉

断开：0 CTRL-P 〈CR〉

例如，要将软盘上的HELLO程序清单打印出来，可按如下步骤进行：

(1) 打入命令LOAD HELLO〈CR〉，装入程序；

(2) 打入命令PR#1 〈CR〉，连通打印机；

(3) 打入命令LIST 〈CR〉，打印机列印程序清单；

(4) 打入命令PR#0〈CR〉，断开打印机。

另外，Ctrl-Reset命令键是强制系列复位，使用了该命令也会强制断开打印机与中华学习机的连通。

PR#命令不但可以在BASIC提示状态下直接连通或断开打印机，而且也可以作为一条语句放在程序中使用。程序一旦执行了PR#1语句，就会连通设在1号槽口上的打印机，输出数据就会由打印机打印出来。同样，执行了PR#0语句

就会解除打印机的输出。

下面这段BASIC程序是在没有DOS操作系统的情况下使用打印机的一个例子。

例 1

```
  5  REM OUTPUT THREE LINES TO  A PRINTER
 10  PR# 1
 20  PRINT TAB (6) ; "Chinese Education Computer"
 30  PRINT
 40  PRINT TAB (12) ; "Made in CHINA"
 50  PR# 0
100  END
```

如果在有DOS操作系统的情况下，连通和断开打印机必须使用DOS命令，上面的例子程序就得改写成下面的格式：

例 2

```
  5  REM OUTPUT THREE LINES TO A PRINTER
 10  PRINT CHR$ (4) ; "PR#1"
 20  PRINT TAB (6) ; "Chinese Educa tion Computer"
 30  PRINT
 40  PRINT TAB (12) ; "Made in CHINA"
 50  PRINT CHR$ (4) ; "PR#0"
100  END
```

7.3.3 设定打印行的宽度

用PR#1命令连通打印机进行程序列表和输出字符串时，可以发现所有列印的程序清单或打印的字符串，最大输出行宽只有40个字符，对于80列打印机来说，正常输出字符每行最多可以打印80个字符，而压缩打印方式每行最多可以打印

132个字符。要改变打印行宽，可以通过置打印行宽的命令来完成。有两种方法可以任意改变打印行宽。

1. 利用软开关设置打印宽度

这是已固化在主机的打印机驱动软件所提供的基本功能。当用PR#1命令接通打印机后，用POKE 1657，n命令便可将打印宽度设置成n字符/行。

例 3 下面程序将打印宽度置成每行50字符。

```
100  REM Sets Column End
110  PRINT CHR$ (4) ; "PR#1"
120  POKE 1657, 50
130  FOR I=1 TO 16
140  PRINT "0123456789";
150  NEXT I
160  PRINT
200  PRINT CHR$ (4) ; "PR#0"

]RUN
01234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789
0123456789
```

注意：

（1）用POKE 1657，n设置的打印宽度超过40字符，打印字符将不显示在屏幕上。

（2）PR#1命令仅在第一次连通打印机时，置打印宽度为40字符，而以后再次连通打印时，并不改变用POKE1657，n命令所设置的打印宽度。

2. 置打印宽度控制命令

这是打印机本身所提供的控制功能。当用PR#1命令连通打印机后，用PRINT语句输出置行宽的控制命令字符：CHR$(27)+“Q”+CHR$(m)，其中m为所设置的打印行宽度。

例 4 下面程序所设置的打印行宽度为50字符/行。

```
100  REM Sets Column End
110  PRINT   CHR$ (4) ; "PR#1"
120  PRINT   CHR$ (27) ; "Q"; CHR$ (50)
130  FOR I=1 TO 16
140  PRINT"0123456789";
150  NEXT I
160  PRINT
200  PRINT CHR$ (4) ; "PR#0"
]RUN

01234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890123456789
0123456789
```

7.3.4 字符串的小写打印

对于早期在Apple Ⅱ上开发的软件，由于不能进行小写字母的输入，因此，输入的都是大写字母。对于给定的一组大写字符串，若要将其在打印机上以小写方式打印出来，可以使用控制字符CHR$ (23) 将这个字符串括起来。

例 5

```
100  REM Print Low Character
110  PRINT   CHR$ (4) ; "PR#1"
120  POKE 1657,80: POKE 33,80:REM Output Printer Only
```

```
130   A$= HELLO": B$="WORLD"
140   PRINT A$; "          "; B$
150   PRINT  CHR$(23); A$; "          "; B$; CHR$ (23)
160   PRINT A$; "          "; B$
170   PRINT
190   POKE 1657, 40: POKE 33, 40: REM With CRT
200   PRINT  CHR$ (4) ; "PR#0"

]RUN
HELLO    WORLD
hello    world
HELLO    WORLD
```

7.3.5 打印字型及打印效果的变化

为了加强文件及报表的打印效果，打印机除了能按正常字符进行输出外，还可以根据需要改变打印字型的大小，改变打印字体，改变打印颜色的深浅浓度等等。

1. 字符的压缩与放大

输出以下的控制字符将改变打印字符的大小。

放大打印	设置	SO	CHR$(14)
	清除	VS	CHR$(20)
压缩打印	设置	SI	CHR$(15)
	清除	DC2	CHR$(18)

注意：

(1) 放大打印命令只对一行有效，当输出回车字符〈CR〉(CHR$(13)) 后，其放大打印方式将自动被清除。压缩打印在执行清除压缩方式前一直为压缩打印状态。

(2) 当压缩打印命令和放大打印命令同时使用时，将会

产生加重打印的效果。

例 6

```
100  REM Printing Mode
110  PRINT  CHR$ (4) ; "PR#1"
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  PRINT   CHR$ (15); "CONDENSED"; CHR$(18);
140  PRINT   TAB(15); "NORMAL";
150  PRINT   TAB(30); CHR$ (14); "ENLARGED"; CHR$
     (20)
160  PRINT   CHR$(15); CHR$(14);
162  PRINT   "CONDENSED+ENLARGED MODE";
164  PRINT   CHR$(18); CHR$(20)
170  PRINT
190  POKE 1657, 40: POKE 33, 40: REM With CRT
200  PRINT   CHR$(4); "PR#0"

]RUN
CONDENSED       NORMAL       ENLARGED
CONDENSED+ENLARGED MODE
```

2. 打印字体的改变

输出CHR$(27)+“4”控制命令后，将使用替代字符集进行打印；输出CHR$(27)+“5”控制命令后，将恢复成用标准字集进行打印。

例 7　替代字符与标准字符的比较。

```
100  REM Alternate Character Set
110  PRINT  CHR$(4); "PR#1"
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  PRINT "Standard"
140  PRINT  CHR$(27); "4";
```

```
142    PRINT "Alternate"
150    PRINT  CHR$(27), "5",
152    PRINT "Standard"
160    PRINT
190    POKE 1657,40: POKE 33, 40: REM With CRT
200    PRINT  CHR$(4); "PR#0"
```

```
]RUN
Standard
Alternate
Standard
```

3．加重打印的使用效果

使用CHR$(27)+“E”控制命令后，将会加重打印效果，使用CHR$(27)+“F”控制命令后，将恢复成一般打印。

例 8 加重打印方式与一般打印方式的效果比较。

```
100   REM Emphasized Mode Cancel
110   PRINT    CHR$ (4) , "PR#1"
120   POKE 1657,80: POKE 33,80: REM Output printer Only
130   PRINT    CHR$ (27) , "E";
140   PRINT   "EMPHASIZED"
150   PRINT    CHR$ (27), "F"
160   PRINT   "PICA-SIZED"
170   PRINT
190   POKE 1657,40: POKE 33,40: REM With CRT
200   PRINT    CHR$ (4) , "PR#0"
```

```
]RUN
EMPHASIZED
PICA-SIZED
```

4．双倍重打方式的改变

为了加深打印字符的颜色，还可以使用双倍重打控制命令。当输出CHR$(27)+"G"控制命令后，后面的输出字符将被重复打印二次，从而达到了加深打印颜色的目的，当输出了CHR$(27)+"H"控制命令后，将恢复成单次打印。

例 9 双倍重打方式与一般打印方式的比较。

```
100  REM Double-strike Mode Cancel
110  PRINT   CHR$ (4), "PR#1"
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  PRINT   CHR$ (27); "G";
140  PRINT  "DOUBLE PRINT"
150  PRINT   CHR$ (27); "H";
160  PRINT  "NORMAL PRINT"
170  PRINT
190  POKE 1657,40: POKE 33,40: REM With CRT
200  PRINT   CHR$ (4); "PR#0"

]RUN
```

DOUBLE PRINT

NORMAL PRINT

除此之外，还有紧凑型字符的打印，多国文字的选择等等控制命令，这里就不一一列举它们的使用方法。在这些命令中，有许多命令可以结合起来使用，使得打印输出的效果多种多样，这需要用户在使用中不断地摸索和提高。

7.3.6 行间距的改变

除了使用打印机内部的DIP开关控制行间距是1/6英寸还是1/8英寸外，也可以通过发送打印控制命令来改变行间距。

行间距的控制命令码及所设置的行间隔量如下：

(1) ESC 0……………………………………1/8英寸

CHR$(27)+CHR$(48)

(2) ESC 1…………………………………… 7/72英寸

CHR$(27)+CHR$(49)

(3) ESC 2……………………………………1/6英寸

CHR$(27)+CHR$(50)

(4) ESC 3 m…………………………………m/216英寸

CHR$(27)+CHR$(51)+CHR$(m)

(1≤m≤255)

(5) ESC A n………………………………… n/72英寸

CHR$(27)+CHR$(65)+CHR$(n)

(1≤n≤85)

对于EPSON九针打印机来说，两根针之间的距离是1/72英寸，针的半径是1/216英寸（见图7-15）。

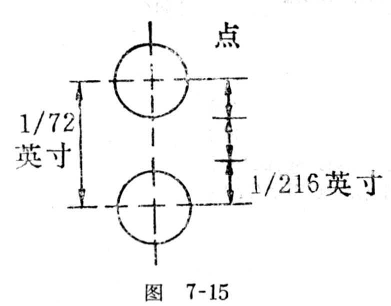


图 7-15

例 10 行间距分别为1/8英寸,7/72英寸及1/6英寸三种情况的比较。

```
100  REM 1/8, 7/72, 1/6 inch Line Spacing
110  PRINT  CHR$ (4) , "PR#1"
```

```
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  PRINT  CHR$(27); "0"; : REM 1/8 inch Line  Spacing
132  FOR I=1 TO 4
134  PRINT"1/8 INCH LINE SPACING"
136  NEXT I
138  PRINT
140  PRINT  CHR$ (27);"1";: REM 7/72 inch Line Spacing
142  FOR I=1 TO 4
144  PRINT"7/72 INCH LINE SPACIN
146  NEXT I
148  PRINT
150  PRINT  CHR$ (27);"2";: REM 1/6 inch Line Spacing
152  FOR I=1 TO 4
154  PRINT"1/6 INCH LINE SPACING"
156  NEXT I
190  POKE 1657, 40: POKE 33, 40: REM With CRT
200  PRINT CHR$ (4), "PR#0"

]RUN

1/8  INCH LINE SPACING
1/8  INCH LINE SPACING
1/8  INCH LINE SPACING
1/8  INCH LINE SPACING

7/72  INCH LINE SPACING
7/72  INCH LINE SPACING
7/72  INCH LINE SPACING
7/72  INCH LINE SPACING

1/6  INCH LINE SPACING

1/6  INCH LINE SPACING

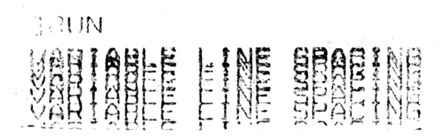
1/6  INCH LINE SPACING

1/6  INCH LINE SPACING
```

例 11 行间距在1/72英寸～12/72英寸之间连续变化。

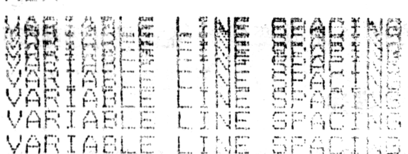
```
100  REM n/72 inch Line Spacing
```

```
110  PRINT  CHR$ (4); "PR#1"
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  FOR I=1 TO 7
140  PRINT  CHR$ (27); "A"; CHR$(I);
150  PRINT"VARIABLE LINE SPACING"
160  NEXT I
190  POKE 1657,40: POKE 33, 40: REM With CRT
200  PRINT  CHR$(4); "PR#0"
```



例 12 本程序的打印效果与例11完全相同。

```
100  REM n/216 inch Line Spacing
110  PRINT  CHR$ (4); "PR#1"
120  POKE 1657,80: POKE 33,80: REM Output Printer Only
130  FOR I=1 TO 10
140  PRINT  CHR$(27); "3"; CHR$ (I * 3);
150  PRINT"VARIABLE LINE SPACING"
160  NEXT I
190  POKE 1657,40: POKE 33,40: REM With CRT
200  PRINT CHR$ (4); "PR#0"
```



7.3.7 设置打印页的长度

对于打印机来说一般认定打印纸的长度是标准的，即11英寸或12英寸一页，当打印机接受到CHR$(12)控制命令时，将走纸到下一页的页首。在实际使用过程中，打印纸的长度是不一致的，我们可以根据需要自行指定打印页的长度。

设置打印页的长度有二种命令：

(1) 按行数设定

CHR$(27)；"C"；CHR$(m)　　　(1≤m≤127)

(2) 按英寸设定

CHR$(27)；"C"；CHR$(0)；CHR$(n)

(1≤n≤22)

7.4 变量的格式打印

变量的格式打印是CEC-ⅠA中华学习机所固化的打印机驱动软件提供的一种辅助功能。

在BASIC程序中，对实型变量的表示方法有其局限性。它不具备有自行指定整数、小数以及指数部分有效位数的功能。因此，给打印及显示数据报表、打印变量表格等带来了一定的困难。打印机驱动软件所提供的这一功能较好地解决了这一困难。用BASIC语言编写程序时，可以通过子程序调用的方法来完成变量的格式显示和打印。

7.4.1 格式的描述

在调用了变量的格式打印子程序后，变量A＝42.517，

则以下列形式表示出来：

(1) 整数表示：In（例如：I3→42）

(2) 定点表示：Fn.m（例如：F4.2→42.51）

(3) 指数表示：En（例如：E3→4.25E+01）

其中，

n：表示数值的有效位数，不包括符号、小数点和指数的长度。n值必须在1～12之间。如果实际数值的有效位数小于指定的n值，则在该数值前面的相应位上填补空格。

m：表示十进制小数点后面的有效位数，$0 \leq m < n$。

指数部分用一位符号和二位数值来表示。

7.4.2 如何调用格式打印子程序

格式打印子程序的入口地址在$C1A0。在调用该子程序之前，首先应将$C1××置为外部I/O地址有效，用PR#1命令启动打印机的接口，如果只需要显示而不打印时，可再使用PR#0命令造成打印机接口的部分断开。在退出格式打印子程序后，若需要完全断开打印机接口，可在执行PR#0命令后，执行POKE 53247,255命令。

对于Apple Ⅱ系统使用EPSON#8132接口板时，其子程序入口地址的计算方法为：4096*12+256*slot，其中slot为该接口板所占用的接口槽号。

1. 命令的调用格式

整数表示：CALL WRITE:X%；I4:

定点表示：CALL WRITE:X；F5.2:

指数表示：CALL WRITE:X；E6:

这里的WRITE是调用子程序的入口地址～49568($C1

A0），它可以是一变量或者是一数值。

在WRITE：与变量之间不应有其它字符。执行"CALL WRITE：："语句时并不送回车换行命令，如果要求换行必须送换行码CHR$(13)，该语句的格式为CALL WRITE：CHR$ (13) ：

2. 命令调用时字符串、空格和变量可以混合出现：

```
CALL  WRITE:J%; I2, " SIN= ", SX; F6.5:
CALL  WRITE:K; I2, "    ", S$, SX; F6.5:
CALL  WRITE:A%(I); I3, "    ", B(I, J, K); F12.8:
```

注意：

(1) 变量与所设定的打印格式之间只能用分号隔开。变量之间，变量与字串之间常用逗号隔开。

(2) 用一个CALL语句输出多个变量时，如果后续变量没有设置变量格式时，按前面的变量格式进行输出。

例如，CALL WRITE：I；I3，" "，X：语句，变量X按格式I3进行显示或打印。

(3) 如果变量的值超过格式所能表示的范围，将输出 ＊＊＊……＊。

7.4.3 格式打印的实例

例 13 下面的程序以三种格式输出函数Y＝EXP(X)的值。Y的三种格式为F8.3，I6，E6。X的格式为I3。格式F8.3的最后一行，因为输出值超过了范围所以输出一行"＊"。

```
100  REM  PRINT FORMATING DEMO
110  WRITE=52480
120  PRINT  CHR$(4); "PR#1":REM
```

```
     INIT FORMATING PROG.
130  PRINT  CHR$ (0), CHR$ (13)
140  PRINT  CHR$ (4), "PR#0":REM
     OUTPUT VIDEO ONLY
150  GOSUB 1000
160  PRINT  CHR$ (4), "PR#1":REM
     OUTPUT PRINTER & CRI
170  GOSUB 1000
180  PRINT  CHR$ (4), "PR#0":REM
     OUTPUT VIDEO ONLY
190  POKE 53247, 255:REM  DEACTIV
     E
200  END
1000  PRINT  TAB (5), "***PRINT F
     ORMATING DEMO ***":PRINT
1010  PRINT  TAB (4), "X", TAB(20), "Y=EXP(X)"
1020  PRINT  TAB (2); "-I3-", TAB(10), "—F8.3—",
     TAB(19); "—I6—", TAB(30), "—E6—":PRINT
1030  FOR I= -7 TO 12
1040  Y=EXP(I)
1050  CALL  WRITE:I, I3, "  ":
1060  CALL  WRITE:Y, F8.3, "  ":
1070  CALL  WRITE: Y, I6; "  ":
1080  CALL  WRITE, Y, E6, CHR$(13):

1090  NEXT
1100  RETURN
```

运行结果如下：

```
*** PRINT FORMATING DEMO ***
```

```
  X                              Y = EXP(X)
—I3—     —F8.3—          —I6—         —E6—

 -7       0.000             0        9.11881E-04
 -6       0.002             0        2.47875E-03
 -5       0.006             0        6.73794E-03
 -4       0.018             0        1.83156E-02
 -3       0.049             0        4.97870E-02
 -2       0.135             0        1.35335E-01
 -1       0.367             0        3.67879E-01
  0       1.000             1        1.00000E+00
  1       2.718             2        2.71828E+00
  2       7.389             7        7.38905E+00
  3      20.085            20        2.00855E+01
  4      54.598            54        5.45981E+01
  5     148.413           148        1.48413E+02
  6     403.428           403        4.03428E+02
  7    1096.633          1096        1.09663E+03
  8    2980.957          2980        2.98095E+03
  9    8103.083          8103        8.10308E+03
 10   22026.465         22026        2.20264E+04
 11   59874.141         59874        5.98741E+04
 12 ***********162754                1.62754E+05
```

例 14 下面的程序是在汉字方式下的格式打印，打印的内容同例1相同。

```
   REM 汉字方式下的变量格式打印程序
10 WRITN=52480
20 POKE-12288, 255: POKE-16127, 0: REM启动格式打印程序
22 POKE 995, 1: REM 仅显示不打印
```

```
25  GOSUB 1000
30  POKE 963,3: REM 显示的同时打印
35  POKE 1659,0: POKE 2043,50: REM 设置打印字型和行宽
40  GOSUB 1000
50  POKE 963,1: REM 关闭打印机
55  POKE 53247,255: REM 断开格式打印程序
100  END
1000  PRINT "TAB(5); "***汉字方式下的格式打印***":
      PRINT
1010  PRINT  TAB(4); "X"; TAB(20); "Y=EXP(X)"
1020  PRINT  TAB(2); "-I3-"; TAB(10); "-F8.3-";
      TAB(19); "-I6-"; TAB(30); "-E6-": PRINT
1030  FOR I= -7 TO 12
1032  Y=EXP(I)
1034  CALL WRITE: I, I3," ":
1036  CALL WRITE: Y, F8.3," ":
1038  CALL WRITE: Y, I6," ":
1040  CALL WRITE: Y, E6, CHR$(13):
1042  NEXT
1050  RETURN
```

```
]RUN
    ***汉字方式的格式下打印***
  X               Y=EXP(X)
—I3—        —F8.3—          —I6—        —E6—
 -7          0.000             0        9.11881E-04
 -6          0.002             0        2.47875E-03
 -5          0.006             0        6.73794E-03
 -4          0.018             0        1.83156E-02
 -3          0.049             0        4.97870E-02
 -2          0.135             0        1.35335E-01
```

-1	0.367	0	3.67879E-01
0	1.000	1	1.00000E+00
1	2.718	2	2.71828E+00
2	7.389	7	7.38905E+00
3	20.085	20	2.00855E+01
4	54.598	54	5.45981E+01
5	148.413	148	1.48413E+02
6	403.428	403	4.03428E+02
7	1096.633	1096	1.09663E+03
8	2980.957	2980	2.98095E+03
9	8103.083	8103	8.10308E+03
10	22026.465	22026	2.20264E+04
11	59874.141	59874	5.98741E+04
12	**********	162754	1.62754E+05

7.4.4 制表格式的打印

在许多情况下需要利用计算机进行表格的处理和打印，如果仅仅使用BASIC语言所提供的TAB(X)函数进行表格打印是不够的，下面介绍两种常用的表格打印的方法。

1. 利用置表格命令

通过向打印机发送命令代码ESC $Dn_1n_2\cdots n_K0$（CHR$(27)；CHR$(4)；CHR$($n_1$)；…CHR$(n_K)；CHR$(0)），在打印机上预置多个TAB定位，然后在每送一组数据之前，送一命令代码Ctrl-I(即CHR$(9))，这样就会按指定的TAB定位次序将多个数据项打印在指定位置上。

值得注意的是：（1）每个数据项之间不得插有回车命令。（2）设置的TAB定位要适当，多个数据项不得侵入到下一个TAB所指定的位置中。

例 15

```
100  REM  PRINTING TABLE
105  PRINT  CHR$(4); "PR#1"
110  PRINT  CHR$(27); "D", CHR$(1);
     CHR$(11); CHR$(21); CHR$(0);
120  FOR I=0 TO 2
130  PRINT  CHR$(9); "ABC",
140  NEXT I
142  PRINT
143  PRINT
150  PRINT  CHR$(4); "PR#0"

]RUN
ABC          ABC          ABC
```

2. 利用控制软开关

1529地址单元是禁止换行软开关，执行POKE 1529, 255命令后，所有输出CR命令代码（即CHR $ (13)）时，都不能使得打印机换行，当执行了POKE 1529, 0命令后，将可以恢复CR换行的功能。

利用这一特点，再结合TAB函数的使用可以在一个输出行上打印多个数据项，以实现制表格式的打印功能。

例 16

```
100  REM  PRINTING  TABLE
200  PRINT  CHR$ (4) ; "PR#1"
300  POKE  1657, 80: POKE  33, 80: REM
       OUTPUT  PRINTER  ONLY
400  PRINT "12345678901234567890l
       234567890l234567890"
500  POKE  1529, 255: REM  DISABLE CR
```

```
600  FOR  I=1  TO  3
700  PRINT  TAB (10  *  I) ; "ABC"
800  NEXT  I
900  POKE  1529, 0: REM  ENABLE  CR
1000  PRINT
1100  POKE  1657, 40: POKE  33, 40: REM
       WITH  CRT
1200  PRINT  CHR$ (4) ; "PR#0"
```

```
]RUN
1234567890123456789012345678901234567890
          ABC       ABC       ABC
```

7.5 位图象的打印

位图象打印命令可以以列为单位进行图象的打印输出，利用这一功能可以在文本打印的同时打印一些特殊符号的形象及划线等等。

7.5.1 位图象打印命令及数据的组成

位图象数据由8位二进制数组成，它们与打印头的每根针的对应关系如图7-16。

常用的位图象打印命令有二个：(1)标准密度的打印ESC K，(2)倍密度的打印ESC L。

为了使得打印机转入位图象打印方式，必须在位图象打印控制命令之后置上有关位图象的数据量参数和位图象数据。其输出顺序如图7-17。

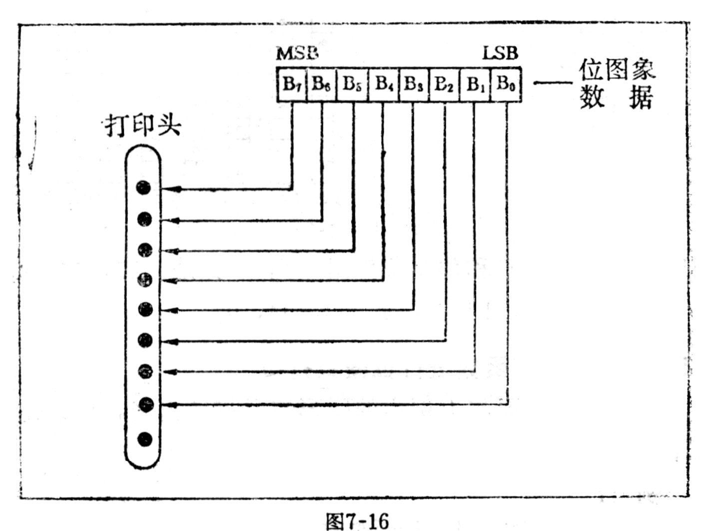


图7-16

ESC代码CHR $ (27)
"K"代码 CHR $ (75)
"L"代码 CHR $ (76)
位图象数据量(字节)除以256所得余数m
位图象数据量(字节)除以256所得商数n
位图象数据

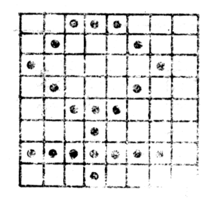
图 7-17

例如，要打印300个字节的位图象，其参数应该是：

m = 300 - INT (300/256) * 256 = 44

n = INT (300/256) = 1

如果我们需要将如图7-18这个图形符号通过打印机打印出来



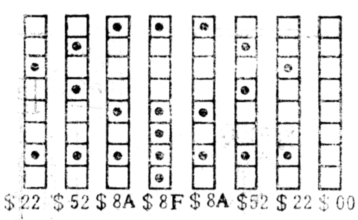


图 7-18

相应的控制命令码和位图象数据序列应该是：27，75，8，0，34，82，138，143，138，82，34，0。

7.5.2 用BASIC函数进行位图象打印

如果我们需要将上例中的图形符号打印出来，可以使用以下这段BASIC程序（参见例17）。

例 17

```
100  REM  Bit-Image Printing-1
110  PRINT  CHR$ (4);"PR#1"
120  POKE 1657,80: REM  Output Pr
     inter Only
130  REM  Send ESC+K+8+0
140  PRINT  CHR$ (27); CHR$ (75);
      CHR$ (8); CHR$ (0);
150  REM   Send Bit-Image Data
160  PRINT  CHR$ (34); CHR$ (82);
      CHR$ (138); CHR$ (143);
170  PRINT  CHR$ (138); CHR$ (82)
     ; CHR$ (34); CHR$ (0);
180  PRINT  CHR$ (13): REM  Send
     CR Code
190  POKE 1657,40: REM  With CRT
200  PRINT  CHR$ (4);"PR#0"

]RUN
```

注意

在例17这段程序执行过程中，由于CHR $(X)函数，X在128～256之间所产生的字符码和X在0～127之间所产生的字符码完全相同。因此，图象的最高位数据没有被打印出来，所以用CHR$(X)函数打印出来的位图象最多可达7个点（最高位被屏蔽）。

7.5.3 用机器语言子程序送数据到打印机

如果使用机器语言子程序将要输出的命令和数据直接送到打印机上，则可避免最高位不能打印的问题。

1、利用CALL语句

(1) 从 $0300地址单元开始建立如下的机器语言子程序：

```
LDA   $00
BIT   $C1C1
BMI   $0302
STA   $C090
RTS
```

(2) 将要输出的控制命令和位图象数据送到 $00单元，再使用CALL 768 语句将数据送往打印机（参见例18）。

例 18

```
100  REM  Bit-Image Printing-2
110  REM  Data Transfer Subroutin
     e
112  RESTORE :AD = 768
114  FOR I = 1 TO 11
116  READ A: POKE AD,A:AD = AD +
```

```
     1
118  NEXT I
120  DATA   165,0,44,193,193,48,2
     51,141,144,192,96
140  REM  SEND ESC+K+8+0
142  POKE 0,27: CALL 768
144  POKE 0,75: CALL 768
146  POKE 0,8: CALL 768
148  POKE 0,0: CALL 768
150  REM  Send Bit-Image Data
152  POKE 0,34: CALL 768
154  POKE 0,82: CALL 768
156  POKE 0,138: CALL 768
158  POKE 0,143: CALL 768
160  POKE 0,138: CALL 768
162  POKE 0,82: CALL 768
164  POKE 0,34: CALL 768
166  POKE 0,0: CALL 768
170  POKE 0,10: REM  Send LF Code

180  END

]RUN
```

2. 利用USR(X)函数

(1) 在 $0A～$0C单元中放入代码 $4C, $00, $30（即JMP $0300指令）。

(2) 从 $0300地址单元开始存放如下的6502机器语言程序：

```
JSR  $E10C
LDA  $A1
BIT  $C1C1
BMI  $0305
STA  $C090
RTS
```

(3) 利用USR(X) 函数将要输出的控制命令和位图象打印数据输出到打印机上。

以下程序将以倍密度图象打印方式重复打印二个前例所示的图形符号。

例 19

```
100  REM  Bit-Image Printing-3
110  REM  Data Transfer Subroutin
     e
112  RESTORE :AD = 768
114  FOR I = 1 TO 14
116  READ A: POKE AD,A:AD = AD +
     1
118  NEXT I
120  DATA  32,12,225,165,161,44,1
     93,193,48,251
122  DATA  141,144,192,96
130  REM  Send JMP Vewctor
132  POKE 10,76: POKE 11,0: POKE
     12,3
140  REM  Send ESC+1+18+0
142 Z =  USR (27) +  USR (76) +  USR
     (18) +  USR (0)
150  REM  Send Bit-Image Data
152  FOR I = 1 TO 18
154  READ A:Z = Z +  USR (A)
```

```
156  NEXT I
160  DATA  0,34,82,138,143,138,82
     ,34,0
162  DATA  0,34,82,138,143,138,82
     ,34,0
170 Z =  USR (10) +  USR (10): REM
      Send LF Code
180  END

]RUN
♀♀
```

7.5.4 位图象打印功能的应用

在这里我们列举二个使用位图象打印功能的实例。

1. 加重打印

把相同的一段文字在一行内重复打印二次，第二次与第一次打印仅相差一个点的距离，从而使得这段文字显得比较突出，取得了加重打印的效果。

例 20

```
5 REM  EMPHASIZED PRINTING
10 PRINT CHR$ (4) ; "PR#1"
20  POKE 1657, 80:REM  OUTPUT  PRI
    NTER ONLY
25  POKE 1529, 255:REM  KILL  LF
30  PRINT"EMPHASIZED  PRINTING"
40  REM  PRINT  ESC+K+1+0, 0
45  PRINT  CHR$ (27) +CHR$ (75)
     +CHR$ (1) +CHR$ (0) +CHR$ (0) ;
50  PRINT"EMPHASIZED  PRINTING"
60  POKE  1529, 0: PRINT: REM  SEN
```

```
   D LF CODE
65 POKE 1657, 40: REM OUTPUT CRT
80 PRINT CHR$ (4) ; "PR#0"
100 END
```

```
]RUN
```

EMPHASIZED PRINTING

2. 下划线打印

利用文本打印和位图象打印的结合，可以取得下划线打印的效果。

例 21

```
5 REM UNDER-LINED PRINTING
10 PRINT CHR$ (4) ; "PR#1"
20 POKE 1657, 80: REM OUTPUT PRINTER ONLY
25 POKE 1529, 255: REM KILL LF
30 PRINT"UNDER-LINED PRINTING"
40 PRINT CHR$ (27) ; CHR$ (75) ; CHR$ (11*6) ;
   CHR$ (0) ;
45 FOR I=1 TO 11 * 6: PRINT CHR$ (1) ; : NEXT
   I:PRINT
50 POKE 1529, 0:PRINT:REM SEND LF CODE
55 POKE 1657, 40: REM OUTPUT CRT
100 PRINT CHR$ (4) ; "PR#0"
```

```
]RUN
UNDER-LINED PRINTING
```

7.5.5 高分辨率图形的硬拷贝

CEC-ⅠA型中华学习机共有三类六个高分辨率图形显示页，它们是：主存第一页、主存第二页、辅存第一页、辅存

第二页、倍高分辨率第一页、倍高分辨率第二页。用户可以根据需要选择对某一类高分辨率图形页进行拷贝打印。

控制高分辨率图形打印的软开关共有三个，它们彼此之间有着一定的联系，它们的地址和作用如表7-2所示。

表 7-2

软开关地址		作用
十进制	十六进制	
1145	$479	选择位图象打印密度
1913	$779	高分辨率图形打印方式寄存器
2042	$7FA	选择图形显示页类型

这些软开关的参数值与作用如下：

(1) 选择位图象打印密度

POKE 1145，75：选择标准密度的位图象打印方式。

POKE 1145，76：选择倍密度的位图象打印方式。

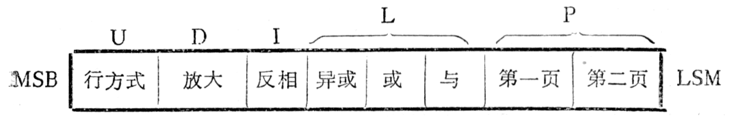
(2) 选择图形显示页类型

POKE 1914，00：选择主存高分辨图形显示页。

POKE 1914，64：选择辅存高分辨图形显示页。

POKE 1914，128：选择倍高分辨图形显示页。

(3) 高分辨图形打印方式寄存器



寄存器各位的含义如下：

(i) 图形页的选择“P”

方式寄存器中的“P”值应和图形显示页类型参数（内

存地址为1914）的值一起使用，它可以确定拷贝哪一个图形显示页。

主存第一页

P=1

（1914）=0

主存第二页

P=2

（1914）=0

辅存第一页

P=1

（1914）=64

辅存第二页

P=2

（1914）=64

倍高分辨第一页

P=1

（1914）=128

倍高分辨第二页

P=2

（1914）=128

当P=0时，将不进行高分辨图形的硬拷贝，当P=3时，可以把由1914地址单元的值所确定的某一类高分辨图形的第一页、第二页并列拷贝到打印机上。

（ii）逻辑操作“L”

所谓逻辑操作是指对高分辨图形或倍高分辨图形的第一页、第二页的图形数据之间进行的逻辑运算。

L＝0　无逻辑操作

L＝4　逻辑与（AND）

L＝8　逻辑或（OR）

L＝16 异或（EOR）

（iii）反相打印“I”

I＝0　正常打印

I＝32 反相打印（将图形数据与$FF异或后再打印）。

（iv）放大打印“D”

D＝0 正常打印

D＝64放大打印（对指定的显示页的图形点水平和垂直方向放大二倍打印）。

（v）行方式硬拷贝

U＝0 整屏硬拷贝

U＝128行拷贝（整个图形显示页由24行组成，按照VTAB的光标定位拷贝相应行的图形。）

这些参数的总和便是方式寄存器的值：

POKE 1913，P＋L＋I＋D＋U

当需要进行高分辨图形硬拷贝打印时，将这些软开关控制单元置上相应的值，然后输出控制命令代码Ctrl-Q（即CHR $（17））即可进行打印。如果需要中止打印，可按Ctrl-C控制命令键（见表7-3）。常用打印机基本控制命令见表7-4。

表 7-3 打印机控制软开关

	软开关地址		控制命令	控制功能
	十进制	十六进制		
1	1145	$479	POKE 1145，75	标准密度图象打印(480点/行)
			POKE 1145，76	倍密度图象打印(960点/行)
2	1273	$4F9	POKE 1273，255	初始化接口单元
3	1401	$579	POKE 1401，YY	YY为小写字母引导符
4	1529	$5F9	POKE 1529，255	禁止执行换行命令(LF)
			POKE 1529，0	释放LF
5	1657	$679	POKE 1657，n	设置打印宽度(n个字符/行)
6	1785	$6F9	POKE 1785，0	清除小写标志
7	1913	$779	POKE 1913，P+L+I+D+U	高分辨图形打印方式寄存器
	2041	$7F9		打印列计数单元
8	2042	$7FA	POKE 1914，n	选择图形显示页类型

表7-4 打印机基本控制命令表

编号	控制命令	Dec.	Hex.	MX-80 II	MX-80III	MP-80III	RX-80III	LX-800	FX-100	命令的控制功能
1	NUL	0	00	●	●	●	●	●	●	ESCB，ESC C等命令的结束标志
2	BEL	7	07	●	●	●	●	●	●	响铃，使打印机鸣叫3秒
3	BS	8	08	●	●	●	●	●	●	打印当前行并退一个字符
4	HT	9	09	●	●	●	●	●	●	水平制表命令的执行
5	LF	10	0A	●	●	●	●	●	●	打印并走纸一行
6	VT	11	0B	●	●	●	●	●	●	垂直制表命令的执行
7	FF	12	0C	●	●	●	●	●	●	打印并走纸到下一页的页首
8	CR	13	0D	●	●	●	●	●	●	打印缓充区的数据
9	SD	14	0E	●	●	●	●	●	●	置放大字型打印方式
10	SI	15	0F	●	●	●	●	●	●	置压缩字型打印方式
11	DO1	17	11	●	○	○	○	●	○	置打印机为数据可接收状态
1	DO2	18	12	●	●	●	●	●	●	清除压缩字型打印方式
备注	● 表示有该项命令功能				○ 表示没有该项命令功能					

表7-4(续)

编号	制控命令	Dec.	Hex.	MX-80Ⅱ	MX-80Ⅲ	MP-80Ⅲ	RX-80Ⅲ	LX-800	FX-800	命令的控制功能
13	DO3	19	13	●	○	○	○	●	○	置打印机为数据不可接收状态
14	DO4	20	14	●	●	●	●	●	●	清除放大字型打印方式
15	CAN	24	18	○	○	○	○	●	○	清除整个打印行的文本数据
16	ESC	27	1B	●	●	●	●	●	●	命令扩展字符在字母数字前
17	ESC SO	14	0E	○	○	○	○	●	○	置放大字型打印方式
18	ESC SI	15	0F	○	○	○	○	●	○	置压缩字型打印方式
19	ESC^	94	27	○	○	○	○	●	○	选择九针图形打印方式
20	ESC!	33	21	○	○	○	○	●	○	各种打印字型的选择
21	ESC%	37	25	○	○	○	○	●	○	设置/清除用户指定的字符组
22	ESC&	38	26	○	○	○	○	●	○	定义用户自己定义的字符
23	ESC*	42	2A	○	○	○	●	●	●	通用位图象打印命令
24	ESC-	45	2D	○	●	●	●	●	●	设置/清除下划线打印方式
备注	● 表示有该项命令功能				○ 表示没有该项命令功能					

表7-4(续)

编号	控制命令	Dec.	Hex.	MX-80Ⅱ	MX-80Ⅲ	MP-80Ⅲ	RX-80Ⅲ	LX-800	FX-100	命令的控制功能
25	ESC 0	48	30	●	●	●	●	●	●	置行间距为1/8英寸
26	ESC 1	49	31	●	●	●	●	●	●	置行间距为7/72英寸
27	ESC 2	50	32	●	●	●	●	●	●	置行间距为1/6英寸
28	ESC 3	51	33	●	●	●	●	●	●	置行间距为n/216英寸
29	ESC 4	52	34	○	○	○	●	●	●	选择替代字符集
30	ESC 5	53	35	○	○	○	●	●	●	选择标准字符集
31	ESC 8	56	38	●	●	●	●	●	●	置无纸检测控制信号无效
32	ESC 9	57	39	●	●	●	●	●	●	置无纸检测控制信号有效
33	ESC:	58	3A	○	○	○	○	●	○	将ROM字符集复制到用户字符区
34	ESC<	60	3C	○	○	○	●	●	●	从左至右地打印当前行
35	ESC@	64	40	○	●	○	●	●	●	初始化打印机
36	ESC A	65	41	●	●	●	●	●	●	设置行间距为n/72英寸
备注	● 表示有该项命令功能　　○ 表示没有该项命令功能									

表7-4(续)

编号	控制命令	Dec.	Hex.	MX-80Ⅱ	MX-80Ⅲ	MP-80Ⅲ	RX-80Ⅲ	LX-800	FX-100	命令的控制功能
37	ESC C	67	43	●	●	●	●	●	●	设置打印纸的长度
38	ESC E	69	45	○	●	●	●	●	●	设置强调打印方式
39	ESC F	70	46	○	●	●	●	●	●	清除强调打印方式
40	ESC G	71	47	○	●	●	●	●	●	设置双倍重打方式
41	ESC H	72	48	○	●	●	●	●	●	清除双倍重打方式
42	ESC J	74	4A	○	●	●	●	●	●	按n/216英寸走纸一行
43	ESC K	75	4B	●	●	●	●	●	●	置正常密度的位图象打印
44	ESC L	76	4C	●	●	●	●	●	●	置倍密度的位图象打印
45	ESC M	77	4D	○	○	○	●	●	●	置高质量字符打印方式
46	ESC N	78	4E	●	●	●	●	●	●	设置打印底边的空白行数
47	ESC O	79	4F	○	●	●	●	●	●	清除ESC N命令功能
48	ESC P	80	50	○	○	○	●	●	●	清除 ESC M 命令功能
备注	● 表示有该项命令功能				○ 表示没有该项命令功能					

表7-4(续)

编号	控制命令	Dec.	Hex.	MX-80Ⅱ	MX-80Ⅲ	MP-80Ⅲ	RX-80Ⅲ	LX-800	FX-100	命令的控制功能
49	ESC Q	81	51	○	●	●	●	●	●	设置打印行宽的列计数
50	ESC R	82	52	○	●	○	●	●	●	选择内部字符集
51	ESC S	83	53	○	●	●	●	●	●	置上下标打印方式
52	ESC T	84	54	○	●	●	●	●	●	清除上下标打印方式
53	ESC U	85	55	○	●	●	●	●	●	设置/清除单向打印方式
54	ESC W	87	57	○	●	●	●	●	●	设置/清除放大打印字型
55	ESC Y	89	59	○	○	○	●	●	●	置倍速倍密度位图象打印
56	ESC Z	90	5A	○	○	○	●	●	●	置四倍密度位图象打印
57	ESC a	96	60	○	○	○	○	●	○	设置打印头的定位位置
58	ESC b	97	61	○	○	○	○	●	○	设置垂直制表的定位位置
59	ESC e	101	65	○	○	○	●	●	●	设置水平/垂直制表单位
60	ESC f	102	66	○	○	○	●	●	●	设置水平/垂直跳格位置
备注	● 表示有该项命令功能					○ 表示没有该项命令功能				

表7-4(续)

编号	控制命令	Dce.	Hex.	MX-80Ⅱ	MX-80Ⅲ	MP-80Ⅲ	RX-80Ⅲ	LX-800	FX-100	命令的控制功能
61	ESC k	106	6A	○	○	○	○	●	○	设置水平/垂直制表的增量
6	ESC 1	108	6C	○	○	○	●	●	●	设置当前打印字符的列位置
63	ESC m	109	6D	○	○	○	●	●	●	选择特殊图形字符集
64	ESC s	115	73	○	○	○	●	●	●	设置/清除半速打印方式
65	ESC t	116	74	○	○	○	○	●	○	选择代码的上半字符集
66	ESC x	120	78	○	○	○	○	●	○	选择简单/印刷体打印方式
67	DEL	127	7F	○	○	○	●	○	●	清除最后一个要打印的字符
备注	● 表示有该项命令功能				○ 表示没有该项命令功能					

第八章　异步串行通信系统

8.1　异步串行通信的基本概念

在计算机应用领域里，常常需要使一台计算机与另一台计算机或终端之间产生通信联系，其目的是通过把一方的数据传给另一方来增加联机双方的信息量。

计算机的通信方式可以是串行的也可是并行的。串行通信是通过一根数据线，每次移动一位二进制位进行通信；并行通信是通过多根数据线每次发送多位二进制位进行通信。虽然并行通信的信息传递速度比串行通信的速度快，但传输线的开销也比串行通信大，尤其在中、远程通信系统中这一矛盾更为突出。此外，串行通信可以利用电传的传输电缆，可借助现有的电话设备，因此能使通信的耗费降低。正因如此，CEC-IA配置了串行通信接口适配器，为中华学习机的用户方便而廉价地组成通信系统提供了必备条件。

在通信过程中，为了可靠地识别数据，要求通信的双方遵守某种约定。在串行通信中这种约定称为异步串行通信协议。异步串行通信协议规定了单根通信线上传输一个信息字符所包含的起始位、数据位、奇偶校验位以及停止位等。

在数据线上无信息传输时，通信线处于逻辑“1”状态，此状态称为识别标记状态；当通信线开始数据传递时，接收的第一位称作起始位，其状态为逻辑“0”；接着起始位

发送的就是数据位。数据位的个数可以选择，其目是在不同的传输环境下使传输效率最佳。此外，数据位发送由最低位开始，由接收方将其还原成原字符数据。为验证传递数据的正确与否，数据位后面可选择奇偶位。如果选择偶校验，则数据位与奇偶位逻辑 1 的个数必须是偶数，否则将出现校验错；反之，若选择奇校验则数据位与奇偶位逻辑 1 的个数要为奇数。在奇偶位之后发送的是停止位，其状态为逻辑 1，其宽度可以选择1位、1.5位或2位。在停止位之后，若有新的数据发送，则它的起始位紧接着上次发送数据的停止位之后，否则，通信线上保持识别标志状态。图8-1是一个信息字符的时序例子，其数据位为8位、奇校验和一位停止位。

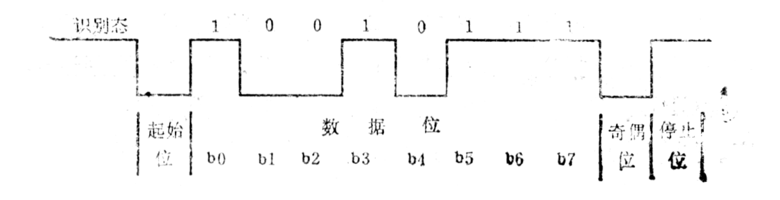


图 8-1　信息字符的例子

通常，串行通信的传输率是以每秒传输的二进制位表示的，度量单位为波特率。每一位二进制的宽度(即传输时间)是波特率的倒数。例如波特率为300则表示一秒钟传输300个二进制位，传输一位所需时间为1／300秒。

综上所述，为保证通信系统的正确性，通信的双方要做到在相同的波特率控制下，并具有同样数目的数据位和停止

位以及相同类型的奇偶校验位。

为了实现上述通信的协议，CEC-ⅠA配置了可编程异步通信接口适配器（ACIA），用户可通过直接对I／O地址的访问，实现异步串行通信，亦可通过系统提供程序调用达到通信的目的。以下几节将对如何实现CEC-IA之间的通信做出详细描述。

8.2 西文方式下的通信方法

在通信之前先要做好通信的准备工作。依照第二章系统操作中关于串行通信的安装方法，将两台CEC-IA型中华学习机用通信连线连接起来，确认安装无误以后，就可以接通这两台学习机电源。

为了便于理解通信接口的工作情况，将CEC-IA划分为三个基本部分描述。首先是基本控制部分，它包括控制设备（如CPU、IOU等）及控制程序（监控程序、BASIC解释程序、通信管理程序等）；其次是基本输入部分，它主要包括键盘；最后是基本输出部分，主要包括CRT显示器、电视机等。

开机之后，通过菜单管理程序选择CEC-BASIC，便进入40列西文文本方式下的BASIC状态，这时，CEC-IA仅以键盘和显示器作为它的输入输出设备，如图8-2所示。

现在，假设甲、乙两台CEC-IA要进行通信，且在某一时刻甲方作为发送方，乙方作为接收方。

当甲乙双方都进入BASIC状态后，在甲方键入BASIC的PR#2命令，在乙方键入IN#2命令，这样甲方便通过2号

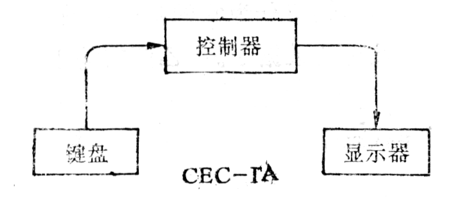


图 8-2 CEC—IA未与外设建立通信时的工作状态

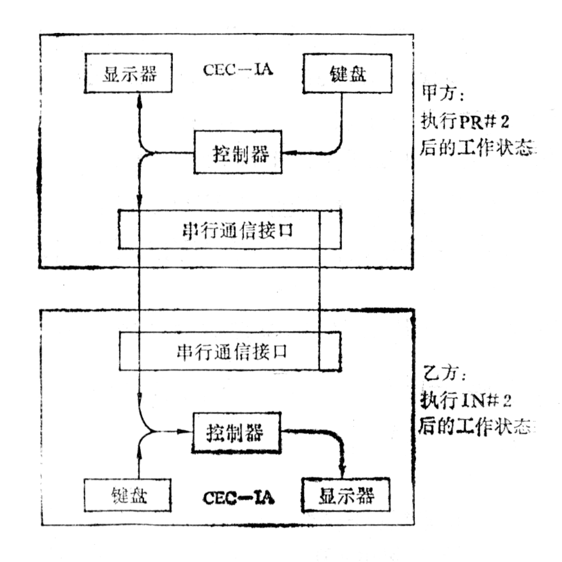


图 8-3 甲乙双方通信时的工作状态

槽口，使串行通信接口适配器作为它的又一个输出设备。同样乙方也把串行通信接口作为它的另一个输入设备。因此，乙方便可收到甲方发送的数据了。甲乙两方的工作情况如图8-3所示。

在这种工作状态下，对于甲方来说，控制器通过键盘接收命令或数据，经处理后送往显示器和通信接口。而乙方的控制器则可同时接收键盘和通信接口送来的数据。

一般情况下，通信要在BASIC语言或机器语言的程序管理下使用。在BASIC语言里，当执行了PR#2之后，再执行输出语句，如PRINT，其输出结果都送给通信接口，再由通信接口发往与之相连的通信设备。当执行了PR#0或强制复位（按Ctrl－Reset键）之后，系统又重新回到基本的输入输出方式下。如果执行了BASIC的IN#2语句后，再执行的输入语句如GET、INPUT等，其输入数据就可从通信接口或键盘得到。但请注意，INPUT语句把所得到的输入字符又发送给输出设备，所以它在输入的同时，也有输出操作。

下面是一个BASIC程序控制通信的例子。这个例子是甲方把“CEC-IA”和$\sqrt{3}$的结果发送给乙方。

甲方BASIC程序：

```
10  PR#2
20  PRINT "CEC-IA"
30  PRINT  SQR (3)
40  PR#0
```

乙方的BASIC程序：

```
10  IN#2
```

```
20 INPUT" "; A$
30 INPUT" "; X
40 IN#0
```

乙方输入程序时，不要在双引号之间键入其它字符，这样，当INPUT执行时就不会出现问号。

首先运行乙方的程序，在甲方程序运行之前，乙方等待输入数据。当运行了甲方程序后，在甲方的显示器上出现

```
CEC-IA
1.73205081
```

这时乙方也收到了上面的结果，并也显示出来，这是由于乙方INPUT语句既接收数据也输出数据；如果不希望在输入的同时又做输出操作，可用GET语句来代替INPUT语句，但GET语句仅接受单字符。

下面这个例子是甲乙双方交互式通信的程序：

甲方：

```
10 IN#2
20 INPUT" "; A$
30 IF A$<>"1" THEN 20
40 IN#0
50 PRINT"PLEASE INPUT:";
60 PR#2
70 INPUT" "; B$
80 PR#0
90 GOTO 10
```

乙方：

```
10 IN#2
20 INPUT" "; A$
30 IF A$<>"2" THEN 20
40 IN#0
```

```
50  PRINT"PLEASE INPUT:";
60  PR#2
70  INPUT " "; B$
80  PR#0
90  GOTO 10
```

分别运行甲乙双方的程序。如果甲方想把自己从键盘得到的数据送到乙方，只需输入"1↵"，这时在显示器上出现

PLEASE INPUT:

这以后再键入的一行字符便可发送给乙方；反之，如果乙方想把键入的字符发给甲方只要输入"2↵"即可。

8.3 汉字方式下的通信方法

通过菜单管理程序，选择CHINESE-BASIC项，便进入了汉字屏幕方式。进入汉字方式时，系统自动执行了PR#3的命令，从而使系统的输入输出程序经3号槽由汉字固化程序管理起来。这时如果仍通过PR#2命令进行通信，则无法在屏幕上得到输出的结果。因此，进入汉字方式后，不能再使用PR#2或IN#2的命令进行通信。

在汉字系统程序内，为用户提供了一个设备控制字单元（$3C3，963）。控制字低4位按排是

bit 0 = {0，字符不送显示器显示；1，字符送显示器显示

bit 1 = {0，字符不送打印机打印；1，字符送打印机打印

bit 2 = {0，字符不送往通信接口；1，字符送往通信接口

bit 3= $\begin{cases} 0，字符不从通信接口输入 \\ 1，字符从通信接口输入 \end{cases}$

在汉字通信时还要注意以下几点：

1。在通信之前要对通信接口的寄存器做初始化，一般调用

CALL 49794

2。由于汉字显示的处理时间较长，所以如果发送方发送得太快，接收方会来不及处理。因此，发送方要把一个汉字的内码分三次发送出去；接收方在接收时要关掉屏幕最下方的提示行（通过输出一个控制字符Ctrl-R可以关掉提示行；再输出一次Ctrl-R又可恢复提示行显示）。

3。为不造成汉字系统的混乱，发送的汉字内码要完整，一般情况下要保证发送的字符个数与接收的字符个数相等。

下面这段BASIC程序是在汉字方式下，甲方将“中华学习机”这5个字发送给乙方。发送完后，甲方只要再按任何一个键又可重新发送一次。

甲方程序：

```
10  CALL 49794:REM 通信初始化
20  POKE 963,5:REM通信端输出
30  A$="中华学习机"
40  FOR  I=1  TO 15
50  X=ASC (MID$ (A$, I, 1) )
60  PRINT CHR$ (X+128) ;
70  NEXT
80  POKE 963, 1:REM 通信端不输出
90  GET  A$
100  PRINT      "再    来    一    次"
```

```
110  GOTO 20
```

乙方程序：

```
5   PRINT CHR$ (18) :REM 关状态行
10  CALL 49794:REM 通信初始化
20  POKE 963, 9:REM 通信端输入
30  FOR  I=1 TO 15:GET B$:
    A$=A$+B$:NEXT
40  POKE 963, 1:REM 通信端不输入
50  PRINT A$
70  A$=" "
80  GOTO 20
```

程序中循环变量终值选择15是因为5个汉字共有3×5个内码。

8.4 通信系统的子程序调用与数据块传递

在槽口ROM区的 $C200～ $C2FF地址范围内，存放着通信系统的固化软件。用户可在机器语言程序里调用它们，调用参数如下：

$47A　从通信口收到的数据

$4FA　待发送的数据。

用户可以调用下述程序：

一、发送子程序（ $C25E）：把要发送的数据存到 $4FA单元，调用此入口，可将数据发送出去。

二、接收子程序（ $C238）：调用此入口，返回时可从 $47A单元得到收到的数据。

三、初始化子程序（$C282）：调用此入口后，可将通信接口寄存器初始化成如下状态：

波特率：9600

时钟：内部产生

数据位：8

停止位：1

奇偶位：奇校验

四、修改信息字符子程序（$C2D4）：此入口的功能在下一节（8.5节）介绍。

五、数据块发送子程序（$C29D）：将待发送数据块的首地址置入 $3C，$3D，末地址置入 $3E、$3F，然后调用此入口即可完成发送数据块的功能。但调用此入口之前需将通信接口寄存器初始化。

六、数据块接收子程序（$C28D）：将要接收数据块的首地址置入 $3C、$3D，末地址置入 $3E、$3F，调用此入口可完成数据块接收功能。调用此入口前要把通信接口寄存器初始化。

用户可根据自己的需要，调用上述系统提供的子程序。调用前需确保通信接口作过初始化，且未按过Ctrl-Reset键。如按过Ctrl-Reset键，需重新调用初始化程序。

在下面这个例子中，甲方将“中华学习机”这几个汉字放大显示在高分辨率图形的第二页，也就是在地址 $4000～$6000一段内，然后将这段地址内的全部数据以数据块的方式发送给乙方；乙方则事先将显示方式置在高分辨率图形的第二页，然后调用数据块接收子程序。

甲方程序：

```
10   POKE DEC ("2FE"), 1:PR#3:PRINT
     REM:进入汉字方式
20   HOME
30   HCOLOR=5
40   A=DEC ("3C5") :REM  汉字显示字形控制字地址
50   POKE A, 31:REM 汉字放大字体
60   PRINT "中华"
70   POKE  A, 19:REM 选另一种字体
80   HCOLOR=2
90   PRINT "学习机"
100  CALL 788:REM 发送
110  END
```

乙方程序：

```
10   HGR2
20   CALL 768:REM 接收
30   END
```

下面这段机器语言子程序是甲乙双方需要的。入口768（$300）是作数据块接收用的；入口788（$314）是作数据块发送用的。

```
0300—   A9 00       LDA   #$00
0302—   85 3C       STA   $3C
0304—   85 3E       STA   $3E
0306—   A9 40       LDA   #$40
0308—   85 3D       STA   $3D
030A—   A9 60       LDA   #$60
030C—   85 3F       STA   $3F
030E—   20 82 C2    JSR   $C282
0311—   4C 8D C2    JMP   $C28D
0314—   A9 00       LDA   #$00
0316—   85 3C       STA   $3C
```

```
0318—    85  3E        STA    $3E
031A—    A9  40        LDA    #$40
031C—    85  3D        STA    $3D
031E—    A9  60        LDA    #$60
0320—    85  3F        STA    $3F
0322—    20  82  C2    JSR    $C282
0325—    4C  9D  C2    JMP    $C29D
```

8.5 通信协议参数的修改

在第一节里，我们已对信息字符的内容作了介绍。通过前一节系统提供的初始化子程序，用户可以得到一个系统指定的信息字符。为了便于用户对信息字符的修改，CEC-IA还提供了一个功能调用。本节将就如何使用这一功能调用说明如下。

首先调用系统初始化子程序 $C282。然后调用 $C2D4（CALL-15660）。此后在屏幕出现

BAUD RATE:50

这表示当前系统处于修改波特率状态。你可通过按空格键选择不同的波特率。每按一次空格，屏幕上就出现一次新的波特率值。当屏幕上出现的波特率是你期望的波特率时，按回车键即可把它作为当前系统指定的波特率。波特率可选择的值有如下9种：

50、75、150、300、600、1200、2400、4800、9600。

选择完波特率后，系统进入数据位选择状态，屏幕上出现

WORD LONG:8

选择方法与波特率选择一样，也由空格键选值，回车键确认。系统提供的可选择的数据位是：

8、7、6、5。

接下去选择停止位，屏幕上出现

STOP BIT:1

系统提供的选择值是

1、2。

选择时请注意：如果数据位选择了5，停止位选择2，则实际上停止位是1.5，而不是2。

最后一项选项是奇偶位，屏幕上出现

PARITY:NO

系统提供可选择的结果是：

NO、ODD、EVEN。

它们分别表示无校验、奇校验、偶校验。

用户可根据通信时设备的环境，指定适当的信息字符，从而确保通信环境在最佳状态下，以提高通信效率。例如，如果传递的所有数据都是5位有效数据，则选择数据位为5就比选择其它数据位效率高。

第九章　音响合成器的使用

9.1 音响合成器的基本功能

中华学习机CEC-IA型增加了音响合成器MSC1。合成器的基本功能如下：

① 具有三个独立的单音频振荡器，可以同时输出三重音。

② 具有谐音发生器，很容易产生音响效果。

③ 单音频振荡器和谐音发生器的音量可以用软件分别控制。

④ 具有包络发生器，可以指定包络图形。

音响合成器MSC1内有14个8位寄存器，上述功能由这些寄存器的输入数值来控制。

9.2 控制寄存器

音响合成器有三个独立的声道。每一个声道都可以单独输出单音或谐音，也可以让单音和谐音同时输出。单音的频率可以独立设置。谐音的频率不能独立设置，三个声道是统一的。单音和谐音的输出受包络的影响。包络可以改变音量的变化规律。包络的周期和图形都可以设置。表9-1为MSC1中寄存器的功能定义。

表 9-1 MSC1寄存器定义

寄存器	功能	7	6	5	4	3	2	1	0	
R_0	A通道频率	低8位数据								
R_1						高4位数据				
R_2	B通道频率	低8位数据								
R_3						高4位数据				
R_4	C通道频率	低8位数据								
R_5						高4位数据				
R_6	谐音频率				5位数据					
R_7	合成口选择	入/出选择		谐音			单音			对应位置1时为off
		口B	口A	C	B	A	C	B	A	置0时为on
R_8	A通道音量				M	4位数据				M＝0时由低四位数据控制音量
R_9	B通道音量				M	4位数据				
R_{10}	C通道音量				M	4位数据				M＝1包络起作用
R_{11}	包络周期	低8位数据								
R_{12}	包络周期	低8位数据								
R_{13}	包络图形					连续	上升	改变	保持	

下面说明这些寄存器的用法。

1. 单音频率控制寄存器R_0～R_5

三个声道的频率由R_0～R_5这6个寄存器的值决定。声道A、B、C对应于R_0与R_1，R_2与R_3，R_4与R_5。它们作为寄存器对来使用，可以写入12位数据。R_1、R_3、R_5的高4位没有意义。以通道A为例，R_0写入低8位数据，R_1的0到3位写入

高4位数据。写入的数据并非频率本身，它与频率的关系如下：

$$f = \frac{f_{clock}}{32 \times D}$$

其中f是输出频率，f_{clock}是MSC1的输入时钟频率，D是写入的十二位数据值。在CEC-IA中输入时钟频率f_{clock}是1MHz。当D取1时，实际输出最高频率为31.25kHz，当D取4095时，输出最低频率为7.6Hz，从而可覆盖全部可听音域。

2．谐音频率控制寄存器R_6

R_6寄存器是用来控制谐音的平均频率的。R_6寄存器是低5位有效，高3位无效。所以R_6的实际有效值是从1到31。

谐音的平均频率f与R_6的写入数值D有如下关系，

$$f = \frac{f_{clock}}{32 \times D}$$

其中f_{clock}是MSC1的输入时钟频率。当D为1时，谐音输出的平均频率为31.25kHz，D为31时，频率为1kHz。

3．混合控制寄存器R_7

R_7是各声道单音、谐音输出的总开关。并且还决定MSC1的二个口的输入与输出的方向。

寄存器R_7的位0到位2的数据决定单音是否输出，位3到位5决定谐音是否输出。各位为0时相当于on，开通输出。为1时相当于off，关闭输出。例如，通道B工作在单音和谐音混合情况时，只要把位1和位4置“0”就可以了。

位6和位7是用来选择输入输出口的。

若确立A口为入口，B口为出口时，必须使位6置“0”且位7置“1”。因此在R_7写入数据时必须注意不使位6和位7变化。

4．声道音量控制寄存器R_8～R_{10}

寄存器R_8～R_{10}分别控制声道A～C的音量。音量由低4位数表示，15时音量最大，0时最小（无声）。但是，当寄存器位4置“1”时，音量就由包络发生器来控制。此时低4位数据无效。当不使用包络发生器时，位4必须置“0”。在任何情况下高三位均是无效的。

5．包络周期控制寄存器R_{11}、R_{12}

包络的周期相当于表9-2中的t。

R_{11}和R_{12}作为寄存器对使用，写入16位数字。R_{11}写入低8位数据，R_{12}写入高8位数据。周期t与16位数值的关系如下：

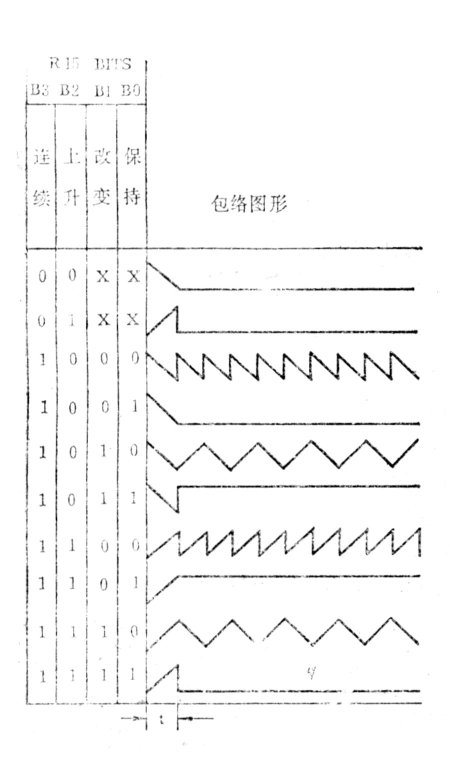
$$t=\frac{512\times D}{f_{clock}}$$

其中f_{clock}是MSC1的输入时钟频率，t是包络周期，D是写入R_{11}、R_{12}的16位数据。

6．包络图形控制寄存器R_{13}

包络图形的选择取决于R_{13}低四位写入的数据。如表9-2中所示。由于包络触发器是在写入数据的同时开始工作的，因而再次工作时，包络图形按指定数据重新修正。

表9-2　包络图形



R15 BITS				
B3	B2	B1	B0	
连续	上升	改变	保持	包络图形
0	0	X	X	
0	1	X	X	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

9.3　音响合成器的编程

音响合成器的编程方法很简单。它一共有三个地址输入口。一个口是指定寄存器号，第二个是写入数据，第三个是

读出数据。如果不改变寄存器号可以连续写入某一个寄存器的数据。

在CEC-IA中，这三个地址口确定如下，

49320（$C0A8）指定寄存器号

49321（$C0A9）写入数据

49322（$C0AA）读数据

例如要给R_0送一个数据可以用下述语句：

```
10  POKE  49320, 0
20  POKE  49321, 99
```

再例如，下面的小程序就是把想好的一批数据顺序送入相应的寄存器得到海浪拍打岸边的音响效果。

```
100  READ A, B
200  IF A = -1 GOTO 800
300  POKE 49320, A
400  POKE 49321, B
500  GOTO 100
600  DATA 6, 5, 10, 31, 12, 30, 11, 0
700  DATA 13, 14, 7, 31, -1, -1
800  END
```

下面的程序例子将同时使用三个声道，一个声道发出海浪的声音，第二个声道发出汽船的声音，第三个声道发出汽笛的声音。通过控制汽船声音的大小，可以得到在海边汽船由远而近，发出汽笛声，然后又驶向远方的音响效果。

```
100  R = DEC ("COA8")
200  N = DEC ("COA9")
300  READ A, B
400  IF A = -1 GOTO 700
410  POKE R, A:POKE N, B:GOTO 300
```

```
500  DATA 6, 5, 10, 31, 12, 30, 11, 0, 13.14
600  DATA 0, 99, 1, 9, 2, 99, 9, 9, 7, 31. -1, -1
700  FOR I=1 TO 9000: NEXT
800  POKE R, 7: POKE N, 18
810  POKE R, 8: POKE N, 6
900  FOR I=1 T0 9000: NEXT
1000 POKE N, 9
1100 FOR I=1 TO 9000: NEXT
1200 POKE R, 7: POKE N, 16
1300 FOR I=1 TO 1000: NEXT
1400 POKE N, 18
1500 FOR I=1 TO 300: NEXT
1600 POKE N, 16
1700 FOR I=1 TO 4000: NEXT
1800 POKE N, 18
1900 FOR I=1 TO 9000: NEXT
2000 POKE R, 8: POKE N, 6
2100 FOR I=1 TO 9000: NEXT
2200 POKE R, 7: POKE N, 31
2300 GOTO 700
]]
```

附录A　BASIC出错信息

在一个错误产生后，BASIC返回到命令一级上，这由提示字符及一个闪烁的光标指出。变量和程序正文仍然是完整的，但是程序不能继续执行，并且所有的GOSUB和FOR循环计数器都置成0，要在一个运行程序中避免这种中断，则可以把ONERR GOTO语句与错误处理例行程序结合起来使用。

当一个错误产生在一个立即执行语句中时，行号不被打印出来。出错信息的格式如下：

立即执行方式的语句　　？XX ERROR

间接执行方式的语句　　？XX ERROR IN YY

在上面的两个例子中，“XX”表示错误名，“YY”是产生错误的语句行号。一个间接执行语句中的错误，只有该语句被执行时才被发觉。

以下是可能产生的出错码以及它们的含义：

CAN'T CONTINUE

试图继续执行一个不存在的程序，或在产生出错后，试图继续执行程序，或者从程序中删除或增加一行之后，试图继续执行程序。

DIVISION BY ZERO

除数为零出错。

ILLEGAL DIRECT

你不能将INPUT,DEF FN,GET,DATA语句作为立即执行命令使用。

ILLEGAL QUANTITY

传送给数学函数或串函数的参数超出范围，产生ILLEGAL QUANTITY错误的情况可能是：

(a)负数为数组下标变量（例如：LET A(-1)=0)。

(b)用负数或零作为LOG的参数。

(c)用负数作SQR的参数。

(d)在A∧B中，A为负数，且B不是一个整数。

(e)在MID$，LEFT$，RIGHT$,WAIT，PEEK，POKE，TAB，SPC,ON…GOTO或作图的任一函数中使用了不恰当的参数。

NEXT WITHOUT FOR

在一个NEXT语句中的变量名与当前的FOR语句中的变量名不一致，或者无变量名的NEXT语句不与任何FOR语句相对应。

OUT OF DATA

执行一个READ语句时，程序中的所有DATA语句早已被读完，程序试图要读更多的数据，但程序中所包含的数据不够。

OUT OF MEMORY

以下的任何一种情况都会产生这种信息：程序太长；变量太多；FOR循环嵌套多于10层的深度；GOSUB的嵌套深度超过24层；表达式太复杂；圆括号嵌套深度多于36层；企图置的LOMEM：值太高；企图置的LOMEM：值低于当前值；企图置的HIMEM：值太低。

FORMULA TOO COMPLEX

执行到多于两条IF“XX”THEN形式的语句。

OVERFLOW

一个计算的结果太长，以致于不能用BASIC的数据格式表示，如果出现下溢出，给出一个零作为结果，并且继续执行，没有任何错误信息打印出来。

REDIM'D ARRAY

对一个数组两次使用DIM语句时，产生此错误。这个错误经常发生在这种情况下：如果直接使用像A(I)=3这样的语句，则BASIC自动定义其标量长度为10，但在程序中又跟有如DIM A(100)的语句。如果你想要找出一个数组定义语句所在的程序行，这个出错信息是很有用的。这只要在第一行上插入该数组的定义语句DIM，运行这个程序，则BASIC将告诉你原来的DIM语句是在什么位置上。

RETURN WITHOUT GOSUB

遇到了一条RETURN语句，但是没有执行相应的GOSUB语句。

STRING TOO LONG

试图通过使用连接运算产生一个多于255个字符的字符串。

BAD SUBSCRIPT

试图访问一个超出数组大小范围的数组元素，如果在一个数组的访问中使用了错误维数，这种出错信息亦产生。例如：当A已经用DIM A(2,2)定义了，但出现了LET A(1,1，1)=Z的语句。

SYNTAX ERROR

在一个表达式中，丢掉了圆括号，在一行中有不合法的字符，不正确的标点符号等等，均会产生此信息。

TYPE MISMATCH

一个赋值语句的左边是一个数值变量，而右边是一个

串，或者左边是一个串变量，而右边是一个数值型数据，另一种情况是一个函数的参数类型是一个串，但给出的却是一个数值型数据，或者反过来。

UNDEF'D STATEMENT

试图使用GOTO、GOSUB使THEN转到一个行号不存在的语句上去。

UNDEF'D FUNCTION

调用了还未定义的用户定义函数。

附录B　BASIC程序存储空间的节省

一、空间的提示

为了使你的程序适合于在配置较小的内存的系统上运行，下面的提示是会有用的。然而，前面两条节省内存空间的方法仅当面对严重的内存紧张的情况下，才予考虑。认真的程序员经常将他的程序保留有两种版本，一种是扩充型的版本，它带有大量文件说明（即带有大量REM）；另一种是“压缩过的”版本，仅占用最少的内存空间。

(1) 每一行上使用多条语句，使得只有少量的空间(5个字节）与程序中的每一行相关连。这五个字节中的两个字节以2进制形式存放了该行的行号。这意味着，在你的行号中不论有多少数字（最小数字是0，最大数字是65529)，它占用同样的字节数（两个）。尽可能地在一行上放入更多的语句将减少你的程序所使用的字节数（一个独立行可包含有239个字符)。

注意：在一行上联合许多语句，使得编辑以及其它修改

都非常困难。这不仅对其它人，对你自己今后回过来看看程序时亦是非常难读和难以理解的。

(2) 删除所有的REM语句，每个REM语句使用的字节数至少为一个字节加上通常的正文的字节数。例如：语句130 REM THIS IS A COMMENT占用了24个字节的内存。语句140 X=X+Y:REM UPDATE SUM中的REM语句占用了12个字节的内存，包括了REM前面的冒号。

注意：和多行程序一样，一个程序没有详述的REM语句，不仅对别人，对你自己今后回过来看程序时亦是非常难读和难于理解的。

(3) 尽可能地使用整型数代替实型数（参看本附录后面的存储分配信息）。

(4) 用变量代替常量。假定你在程序中使用常量3.14159十次，如果你在程序中插入一条语句10 P=3.14159，并在每次需要用时，以P代替3.14159，你将节省40个字节，这亦将引起速度的提高。

(5) 一个程序不需要用END来终止，因此，在程序的末端的END语句可被删除。

(6) 重复使用相同的变量。如果你有一个临时变量T，用于在一部分程序中保留一个临时结果，则在程序的后面部分中又需要临时变量时，可再次使用它。如果你要求计算机的用户在程序执行期间，在两个不同的时间上，用YES或NO来回答两个不同的问题，则可使用同样的临时变量A$存储这个回答。

(7) 用GOSUB执行完成相同作用的程序语句段。

(8) 使用矩阵的零元素。例如：A(0),B(0,X)

(9) 当A$="CAT"被赋值为A$="DOG"时，老的串"CAT"不从存储器中抹去，在你的程序中定期地使用形如X=FRE(0)的语句，将使CEC-BASIC从存储的顶部开始"内部清洗"老的字串。

二、存储的分配

简单的（非数组）实型、整型或串变量，象V，V%或V$使用7个字节。实型变量用2个字节作为变量名，用5个字节作为其值（1个指数，4个尾数）；整数变量用2个字节作为变量名，2个字节作为其值，其余3个字节为0；串变量用2个字节作为变量名，1个字节为串的长度，2个字节为指向内存中串的位置的指针，其余2个字节为0。请参见CEC-BASIC变量分配图。

实型数组变量最少使用12个字节：两个字节为变量名，两个字节为数组的大小，一个为维数，每一维的大小使用两个字节，每个数组元素使用五个字节。整型数组变量的每个数组元素使用两个字节。串数组变量的每个元素使用三个字节：一个字节作为长度，二个字节作为指针，见CEC-BASIC变量分配图。

串变量，无论是简单的还是数组，对串中每个字符使用内存的一个字节。串本身是按照在程序中的先后次序来定义的，从HIMEM:开始存放。

当使用一个DEF语句定义了一个新的函数时，6个字节被用来存储指向定义的指针。

保留字如FOR，GOTO或NOT以及内部函数的名，如COS，INT以及STR $仅占据程序存储的一个字节，程序

中所有其它字符，每个字符占用一个字节。

当一个程序开始执行时，在栈中空间是按下列规则动态分配的：

（1）每个有效的FOR…NEXT循环用16个字节。

（2）每个有效的GOSUB（还没有执行到RETURN）用6个字节。

（3）对一个表达式中的每对圆括号用4个字节，以及表达式中的每个计算出的结果用12个字节。

附录C　提高BASIC程序执行速度

下面的提示将改进你的BASIC程序的执行时间。注意：其中某些提示同样可用来减少你的程序所占用的内存空间。这意味着在许多情况下，你可以提高你的程序执行速度。同时，你还可以改进内存的使用效率。

（1）这大概是最重要的速度提示（速度可提高十分之一）：使用变量代替常量。把一个常量转换成它的浮点（实型数）表示要比读取一个简单变量或数组变量花费的时间多，尤其是在FOR…NEXT循环内部或在其它反复被执行的编码中这是非常重要的。

（2）在BASIC程序的执行期间，首先遇到的变量是从变量表的初始位置开始分配的变量。这意思是，例如一个语句5 A=0:B=A:C=A在变量表中第一个位置放A，第二个放B，第三个放C（假定第5行是程序中一个执行的语句）。在后面的程序中，当BASIC碰到一个引用变量A时，它将只搜索变量表的第一项就找到了A，搜索第二项

可找到B，搜索第三项才找到C等等。

(3) 使用不带下标变量的NEXT语句，NEXT比NEX-TI稍微快一点，因为，不要检查在NEXT中指出的变量是否与最近的仍然有效的FOR语句中变量一样。

(4) 程序执行期间，当BASIC遇到一个新行引用时，例如："GOTO 1000"它将从起始行开始扫描整个用户程序，直到找出所要引用的行号为止(在这个例子中是1000)，因此，经常引用的行在程序中要尽可能地放前一些。

附录D　BASIC保留字及内码的10进制值

一、对照表

10进制值	保留字	10进制值	保留字
128	END	142	HLIN
129	FOR	143	VLIN
130	NEXT	144	HGR2
131	DATA	145	HGR
132	INPUT	146	HCOLOR=
133	DEL	147	HPLOT
134	DIM	148	DRAW
135	READ	149	XDRAW
136	GR	150	HTAB
137	TEXT	151	HOME
138	PR#	152	ROT=
139	IN#	153	SCALE=
140	CALL	154	MNT
141	PLOT	155	TRACE

10进制值	保留字	10进制值	保留字
156	NOTRACE	185	POKE
157	NORMAL	186	PRINT
158	INVERSE	187	CONT
159	FLASH	188	LIST
160	COLOR=	189	CLEAR
161	POP	190	GET
162	VTAB	191	NEW
163	HIMEM:	192	TAB (
164	LOMEM:	193	TO
165	ONERR	194	FN
166	RESUME	195	SPC (
167	RECALL	196	THEN
168	STORE	197	AT
169	SPEED=	198	NOT
170	LET	199	STEP
171	GOTO	200	+
172	RUN	201	−
173	IF	202	*
174	RESTORE	203	/
175	&	204	∧
176	GOSUB	205	AND
177	RETURN	206	OR
178	REM	207	>
179	STOP	208	=
180	ON	209	<
181	WAIT	210	SGN
182	LOAD	211	INT
183	SAVE	212	ABS
184	DEF	213	USR

10进制值	保留字	10进制值	保留字
214	FRE	229	VAL
215	SCRN(	230	ASC
216	PDL	231	CHR$
217	POS	232	LEFT$
218	SQR	233	RIGHT$
219	RND	234	MID$
220	LOG	235	MUSIC
221	EXP	236	PLAY
222	COS	237	SASM
223	SIN	238	BCLR
224	TAN	239	DEC
225	ATN	240	HEX$
226	PEEK		
227	LEN		
228	STR$		

二、保留字

&

ABS AND ASC AT ATN BCLR

CALL CHR$ CLEAR COLOR= CONT

COS

DATA DEC DEF DEL DIM DRAW

END EXP

FLASH FN FOR FRE

GET GOSUB GOTO GR

HEX$ HCOLOR= HGR HGR2 HIMEM:HLIN

HOME HPLOT HTAB

IF IN# INPUT INT INVERSE

LEFT$ LEN LET LIST LOAD

LOG LOMEM: LG

MID$ MNT MUSIC

NEW NEXT NORMAL NOT NOTRACE

ON ONERR OR

PDL PEEK PLOT POKE POP

POS PRINT PR# PLAY

READ RECALL REM RESTORE RESUME

RETURN RIGHT$

RND ROT= RUN

SASM SAVE SCALE=SCRNC SGN SHLOAD

SIN SPC(

SPEED= SQR STEP STOP

STORE STR$

TAB(TAN TEXT THEN TO

TRACE

USR

VAL VLIN VTAB

WAIT

XPLOT XDRAW

BASIC“能辨认出”的这些保留字，每个字仅占用程序存储的一个字节，所有其它的字符在程序存储器中各占用一个程序存储字节。见附录D的保留字值。

连接符（&）打算仅作为计算机内部使用，它不是一个正规的BASIC命令，这个符号当作为一条指令执行时，使其无条件地转到$3F5的位置上。

XPLOT是一个保留字，不作为当前的一个BASIC命令。

对于某些保留字，CEC-BASIC要根据其上下文才能识别：

COLCR, HCOLOR, SCALE, SPEED和ROT

仅当下一个非空格字符是赋值符号"="时，它们才作为保留字，在COLOR和HCOLOR的情况下，保留字OR会妨碍它们用作任何变量名。

SCRN SPC和TAB

仅当下一个非空格字符在圆括号中时，它们才作为保留字。

HIMEM：如果它作为一个保留字，那么，也必须有一个冒号（:）。

ATN仅当在T和N之间没有空格，它才作为保留字，如果在T和N之间存在一个空格，那么AT做为一保留字，而不是ATN。

TO除非前面有A并且在T和O之间无空格，否则把它们作为保留字，如果T和O之间存在一空格，那么把AT作为保留字，而不是把TO作为保留字。

有时圆括号可用来回避保留字：

```
100  FOR A=LOFT OR CAT TO 15
```

用LIST列出来为：

```
100  FOR A=LOF TO RC ATTO 15
```

但是100 FOR A=(LOFT) OR (CAT)TO 15

用LIST列出来，则仍然是：

```
100  FOR A= (LOFT) OR (CAT) TO 15
```

附录E PEEK、POKE和CALL的用法

这儿是一些你以借助于PEEK，POKE或CALL的命

令来使用BASIC的特殊性能。注意：其中某些与BASIC中其它一些命令有完全相同的效果。

简单的开关动作是依赖于地址的，任何包含那个地址的命令对那个开关将起作用。例如：POKE-16304，0就是一个例子，对于这个例子，你可以通过使用POKE来把从0到255中的任何数放到那个地址中，也可通过PEEK来对那个地址进行读取：X＝PEEK(－16304)来获得相同的效果。

这对某些命令不适用。在这些命令中，你必须使用POKE来把一个特定的值放到所需的地址中，这些特定的值可以置一个页边的空白，或者用来把光标移到一个指定的地方。

一、置文本显示窗口

在下面例子中行号为10，20，30和40的前四个POKE命令是用于置TV屏幕上的“窗口”大小。在这个窗口上，显示正文，并且能够自动“上滚”。这些命令分别地置窗口的左边空白，行的宽度，窗口的顶部空白和底部空白。

10 POKE 32，L

置左边的空白，其大小由L指出的值确定，值的范围在0到39之间，这儿的0是指最左边的位置。

窗口的宽度不是由这条命令来改变的，这意味着右边的空白是根据你移动左边空白的同样数目来变化。为了保护你的程序及BASIC，首先应适当压缩窗口的宽度，然后改换左边空白。

20 POKE 33，W

TV显示的宽度（每行字符的个数）为W指出的值，这

个值在1到40的范围内。

不能置W为零：POKE33,0会破坏BASIC。

如果W小于33,PRINT命令的第三个制表（TAB）域可能把字符打印在窗口外。

30 POKE 34，T

根据T指定的值建立TV显示的顶部空白，其T的范围在0到23之间，这儿的0是屏幕的顶行。POKE 34，4将不允许在屏幕的前四行上打印文本。所置的窗口的顶部的空白(T)不能低于底部的空白（下面的B)。

40 POKE 35，B

根据B指定的值建立屏幕底部空白，其B的范围是0到24之间。这儿的24是屏幕底部的一行，窗口底部的空白（B)不能高于顶部空白（即上面的T)。

二、有关文本显示、文本显示窗口和键盘的一些命令

45 CALL—936

清除文本显示窗口里面的所有字符，且移动光标到窗口的左上角。

50 CALL—958

清除在文本窗口里从当前光标位置到底部空白处的所有字符，在当前行上光标上面的字符以及光标左边的字符将不受影响。

60 CALL—868

清除当前行从光标位置到右边空白。

70 CALL—922

发出一个换行。

80 CALL—912

上滚正文一行，即在所定义的窗口内的每行正文都向上移动一行位置，老的顶端的行丢失。老的第二行成为第一行。底端的一行现在是空白。在所定义的窗口外的字符不受影响。

90 X=PEEK（—16384）

读键盘，如果X>127那么表示已按了一个键，X的内容是所按键的ASCII码的值，且第7位为1，这在长程序中是有用的。在这个程序中，计算机检查用户是否想要用新数据去中断一下程序，而不是停止程序的执行。

100 POKE—16368，0

复位键盘选通，以便下一个字符可以读入，这在读入键盘后应立即去做。

三、与光标有关的命令

110 CH=PEEK（36）

读回当前光标的水平位置，并且把这个值赋给变量CH。光标将在0到39的范围内，且是与文本显示窗口的左边空白有关的光标的位置。这个左边空白是由POKE32,L置的，因此，如果由POKE32,5置了左边空白，那么，窗口中最左边的字符是在从屏幕的左端起的第六个打印位置上，且如果PEEK(36)送回的值是5，那么，光标是在从屏幕的左边缘起第十一个打印位置上，即在从文本显示窗口的左起第六个打印位置上（最初听起来有些混乱，因为最左边的位置是0，而不是1），这与POS(X)函数是相同的（参看下一个例子）。

120 POKE 36, CH

移动光标到从本文显示窗口的左起第CH+1个打印位置上（例：POKE 36, 0 将使得下一个字符从窗口的左边打印）。如果窗口左边空白在用PRINT打印之前必须改变，CH必须小于或等于由POKE 22, W置的窗口的宽度，且必须大于或等于0。象HTAB一样，这命令能够把光标移出正文窗口的右边空白，但仅够打印一个字符。

130 CV=PEEK (37)

读出当前光标的垂直位置，且置CV等于该值，CV是光标的绝对垂直位置，它与正文屏幕的顶部或底部的空白无关。因此CV=0是屏幕的顶端行，CV=23是底行。

140 POKE 37, CV

移动光标到由CV指定的绝对垂直位置上。0是最上面的一行，23是底行。

四、有关作图的命令

为了显示文本和制图，中华学习机的内存划分出四个区域：文本显示的第1页和第2页以及高分辨率图形显示的第1页和第2页。

（1）本文显示的第1页是所有的文本和低分辨制图的通用的内存区域，这通过TEXT和GR命令来使用。

（2）在内存中文本显示的第2页刚好位于文本显示的第1页的上面。对用户来讲，这页是不容易存取到的。同正文第1页一样，存储在正文第2页上的信息可以解释为正文，也可以解释为低分辨率图形，或者两者皆可。

（3）高分辨率图形的第一页驻留在中华学习机的内存

的8K到16K上，这个区域是通过HGR命令来使用的。

（4）高分辨率图形的第2页驻留在中华学习机内存的16K到24K上。这个区域是通过HGR2命令使用的。

你可以用BASIC的正文和制图命令或者用这四种不同的开关来使用不同的制图和正文方式。如同这里讨论的许多开关一样。PEEK或POKE可对一个地址置开关为一种状态，以及PEEK和POKE可对第2个地址置开关为另一个状态，简单地说，这四种开关的选择区间为：

(1)文本显示 (POKE－16303，0)

高分辨率或低分辨率图形显示 (POKE－16304，0)

(2)文本或高分辨率显示的第一页(POKE－16300，0)

文本或高分辨率显示的第二页 (POKE－16299，0)

(3)文本显示的第一页或制图的第二页

(POKE－16293，0)

制图的高分辨率的第一页或第二页

(POKE－16297，0)

（4）满屏幕高分辨率或低分辨率制图

(POKE－16302，0)

高分辨率或低分辨率制图加文本显示混合方式

(POKE－16301，0)

```
150 POKE—16304，0
```

从文本显示方式转变到彩色制图方式中，而不清除制图屏幕为黑色。根据另外三个开关的置位，转变成制图方式，可以是低分辨率或高分辨率（从第1页到第2页），也可以是制图加文本显示的混合方式或满屏幕制图方式。

BASIC中类似的命令：GR命令用来转变成第一页的低分辨率制图加正文的混合屏幕，并且清除制图屏幕为黑色；HGR命令用来转变成为第一页高分辨率制图加文本显示的混合屏幕，并且清图形屏幕为黑色；HGR2命令用来转变成第二页高分辨率满屏图形且清除整个屏幕为黑色。

160 POKE—16303，0

从任何彩色制图显示方式转变成所有文本显示方式，而不重置“上滚”窗口，根据第一页/第二页开关的置位，可将文本显示页从第一页转换成第二页，或从第二页转换成第一页。

TEXT命令将屏幕置成全文本显示方式，它除了选择文本显示的第一页以外，还重新将窗口复位成最大，并将光标移到显示屏幕的左下角上。

170 POKE—16302，0

转换制图加文本显示的混合屏幕为满屏幕制图显示方式。

根据其它开关的置位，可以作为正文或是一个40乘48的栅格上的低分辨率制图，或一个278乘192栅格的高分辨率制图出现。

180 POKE—16301，0

从满屏图形转换成图形加文本显示的混合屏幕方式，在屏幕的底部有4行文本显示区，每行40个字符。

根据其它开关的置位，屏幕的上部分可以显示文本，也可以在一个40乘48的栅格上进行低分辨率制图或在一个278乘160栅格上进行高分辨率制图。屏幕的两部分显示将来自同样的页数（1或2）。

184 POKE—16300，0

从第二页转换到第一页，不清除屏幕或移动光标。当你从BASIC进入到整型BASIC时这是必须的，否则，你看到的将仍然是内存的第二页内容。

根据其它开关的置位，这可以使得显示从高分辨率制图的第二页改变到高分辨率制图的第一页，从低分辨率制图的第二页转变为低分辨率制图的第一页，或者从文本显示的第二页转变成文本显示的第一页。

186 POKE 16299，0

从第一页转换到第二页，不清除屏幕或移动光标。

根据其它开关的置位，这可以使得显示从高分辨率制图的第一页改变成高分辨率制图的第二页；从低分辨率制图的第一页改变成低分辨率制图的第二页，或者从正文第一页改变成正文第二页。

190 POKE—16298，0

改变制图页，从一个高分辨率制图的页转变到正文的同样页上，不清除屏幕。当你从BASIC进入到整型BASIC时，这是必须的。否则，整型BASIC的GR指令可能错误地显示高分辨率页。

依赖于其它开关的置位，这可以使得显示从高分辨率制图的第一页改变为低分辨率制图的第一页；从高分辨率制图的第二页改变到低分辨率制图的第二页；或者（在正文方式下）可以使得显示不改变。

195 POKE—16297，0

对图形换页，从一个文本显示页到高分辨率的相应页，不清除屏幕。

依赖于其它开关的置位，这可以使得显示从低分辨率图

形第一页变换到高分辨率图形的第一页，从低分辨率图形的第二页变换到高分辨率图形的第二页，或者（在正文方式下）可以使显示不改变。

200 CALL—1994

清除文本显示的第一页的上面20行为保留的符号@，如果你是在第一页低分辨率制图方式中，则清除制图屏幕的上面40行为黑色，这对正文第二页或对高分辨制图不起作用。

205 CALL—1998

清除整个文本显示的第一页为保留的符号@，如果你是在第一页低分辨率满屏制图方式中，则清除整个屏幕为黑色。这对正文第二页或对高分辨率制图不起作用。

200 CALL 62450

清除当前高分辨率屏幕为黑（BASIC记住上次使用的屏幕，而不考虑开关的置位）。

210 CALL 62454

清除当前高分辨率屏幕为最近绘制使用的高分辨率的颜色（HCOLOR）（BASIC记住最后使用的屏幕，而不考虑开关的置位）。在这前面必须有一个HPLOT语句。

五、与游戏控制器及扬声器有关的命令

220 X=PEEK（-16336）

触发（TOGGLE）扬声器一次，扬声器发出一个“卡嗒”声。

225 X=PEEK（-16352）

触发盒式录音带输入一次，在盒式录音机上产生一个“卡嗒”声。

230 X=PEEK (-16287)

读#0游戏控制器上的按纽开关，如X>127，那么按纽已被按了。

240 X=PEEK (-16286)

同上面的一样，只是读#1游戏控制器的按纽开关。

250 X=PEEK (-16285)

读#2游戏控制器上的按纽。

六、与出错有关的命令

340 X=PEEK (218) +PEEK (219) *256

如果一条ONERR GOTO语句已执行过，这条语句置X等于产生错误的那条语句行号。

350 IF PEEK (216) >127 THEN GOTO 2000

如果内存222单元 (ERRFLG) 的第7位已经置为真，那么，已经遇到一条ONERR GOTO语句。

360 POKE 216, 0

清除ERRFLG，以便发出标准出错信息。

370 Y=PEEK (222)

置变量Y为描述出错类型的编码，这个错误已使得一条ONERR GOTO转移出现。错误类型描述见ONERR GOTO语句。

附录F BASIC零页的使用

单元位置 (16进制)	说明
\$0—\$5	继续CEC-BASIC的转移指令，(CEC-BASIC的RESET 0G RETURN等于整型BASIC的RESET CTRL-CRETURN)。

单元位置（16进制）	说　　　　明
$A—$C	USR函数的转移指令地址见USR函数描述。
$D—$17	通常用于CEC-BASIC的计数器／标志。
$20—$4F	系统监控的保留单元。
$50—$61	通常用作CEC-BASIC的指针。
$62—$66	上次乘法／除法的结果。
$67—$68	指向程序开始的指针，对ROM版本来讲通常置成$0801，或对RAM（盒式录音带）版本来讲通常置成$3001。
$69—$6A	指向简单变量区域开始位置的指针，亦指向程序的结束，并加上1或加上2，除非用LOMEM:语句来修改。
$6B—$6C	指向数组区域开始位置的指针。
$6D—$6E	指向使用中的数值存储区域的末端的指针。
$6F—$70	指向串存储区域开始的指针，串可从这儿开始一直存储到内存的末端。
$71—$72	通用指针。
$73—$74	CEC-BASIC可用的内存的最高单元加1。在最初进入CEC-BASIC时，置成最高的可用的RAM内存地址。
$75—$76	当前正在执行的程序行的行号。
$77—$78	“旧行号”通过CTRL-C，STOP或END语句来建立的，给出程序执行时被中断的行的行号。
$79—$7A	“旧的正文指针”指出下一次将要执行的语句在存储器中的地址。
$7B—$7C	当前DATA语句的行号，该行中的数据正由READ语句来读。
$7D—$7E	指出内存中的绝对单元地址，READ语句正在从这个单元地址中读DATA中的数据。
$7F—$80	指向当前输入的源程序的指针，在INPUT语句执

单元位置（16进制）	说　　明
	行期间置为$201，在一条READ语句执行期间被置成在程序中DATA是从中读入的单元。
$81—$82	保存上次使用的变量名。
$83—$84	指向上次使用的变量的值的指针。
$85—$9C	通用。
$9D—$A3	主要的浮点累加器。
$A4	通常用于浮点的数学例行程序。
$A5—$AB	次要的浮点累加器。
$AC—$AE	通常用于标志／指针。
$AF—$B0	指向程序的末尾（不能用LOMEM:改变）。
$B1—$C8	CHRGET例行程序。CEC-BASIC每次希望得到另一个字符时，在这儿调用它。
$B8—$B9	指向通过CHRGET例行程序得到的上一个字符的指针。
$C9—$CD	随机数。
$D0—$D5	高分辨率制图划线（Scratch）指针
$D6	标志单元
$D8—$DF	ONERR指针／划线。
$E0—$E2	高分辨率制图的X和Y坐标。
$E4	高分辨率制图的颜色字节。
$E5—$E7	通常用于高分辨率制图。
$E8—$E9	指向图形表开头的指针。
$EA	高分辨率制图的碰撞计数器。
$F0—$F3	通用标志
$F4—$F8	ONERR指针。

附录G　BASIC命令参考卡

1. 简单变量

类型	名	范围
实型	AB	+／-9.99999999E+37
整型	AB%	+／-32767
串	AB$	0到255个字符

这里A是字母，B可以是字母也可以是数字。名可以由两个以上的字符组成，但仅有最前面的二个字符是有意义的：AB%和AB3QS%是同一整数变量。

2. 数组变量

类型	典型的元素名
实型	AB（3，12，7）
整型	AB%（3，12，7）
串	AB$（3，12，7）

数组的大小是受有效内存所限制的。

3. 代数运算符

=	赋值给变量（LET是选用项）
-	负号
∧	次幂
*	乘
／	除
+	加
-	减

4. 关系和逻辑运算

=	等于
<>	不等
<	小于

<=	小于或等于
>	大于
>=	大于或等于
NOT	逻辑“非”
AND	逻辑“与”
OR	逻辑“或”

关系和逻辑表达式为真时，其值为1，为假时，其值为0，逻辑运算符也能够用于串的比较。

5. 系统和实用命令

LOAD	从磁带上装入程序。
SAVE	把程序存到磁带上。
NEW	删除当前程序。
RUN	从最小行号处开始运行程序。
RUN477	从477行开始运行程序。
MNT “L”	执行监控反汇编命令
PLAY	装入并执行磁带游戏。
MUSIC X，Y	音乐语句，X为频率，Y为时间。
SASM	进入小汇编。
STOP	停止执行并报告停止那一行。
END	停止执行，但不显示信息。
CTRL—C	用于立即执行方式来停止程序或列表。
RESET	无条件地转向监控程序。用CTR L-C或0G来返回到CEC-BASIC。
CONT	继续执行由STOP，END或CTRL-C所终止的程序。
TRACE	提供调试手段，列出它每次新执行到的每一行的行号。
NOTRACE	关闭TRACE。
PEEK (X)	送回存储在单元X的内容。
POKE X，13	将内存单元X的内容修改成13。
WAIT X，Y，Z	等待，直到X单元的内容与Z进行XOR运算，并

	且再与Y进行AND运算。所得到的值为非零时为止。
CALL X	转向以内存单元X开始的机器语言子程序。
USR (X)	把值X送给机器语言子程序。
HIMEM:	置CEC-BASIC程序可使用的最高的内存有效地址。
LOMEM:	置CEC-BASIC程序可使用的最低的内存有效地址。

6. 编辑命令和与格式有关的命令

LIST	列出整个程序。
LIST X-Y	列出从X行到Y行的程序段。
DEL X，Y	删除从X行到Y行的程序段。
REM XYZ	用来书写程序的注解，程序将这部分忽略。
VTAB Y	把光标移到Y行（1到24）。
HTAB X	把光标移X位置上（1到40）。
TAB (X)	仅用于PRINT语句，它把光标移到X位置上（1到40）。
POS (0)	送回光标当前所在的水平位置的值（0到39）。
SPC (X)	仅用于PRINT语句，在最后一次打印的项和要打印的下一项之间放上X个空格。
HOME	清除屏幕并把光标放在顶部。
CLEAR	所有的变量重置为零。
FRE (0)	送回用户还可使用的内存总数。
FLASH	置计算机的输出为闪烁方式。
INVERSE	置计算机的输出为白底黑字方式。
NORMAL	关闭闪烁或白底黑字方式。
SPEED=X	置字符输出速率为X（从0到255）
ESC A	把光标向右移一格。

ESC B	把光标向左移一格。
ESC C	把光标向下移一格。
ESC D	把光标向上移一格。
右箭头（→）	把光标下的字符输入内存，并把光标向右移一格。
左箭头（←）	删除当前行中刚刚打入的字符，并把光标向左移一格。
CTRL—X	删除当前打入的行。

7. 数组和串

DIM A（X，Y，Z）	置A的最大下标，保留X+1 * Y +1 * Z+1个实型元素的存储空间。由A（0，0，0）元素开始。
DIM A $（X，Y）	置A$的最大下标，它可以包含X+1 * Y+1个串元素，每个元素最多可达255个字符。
LEN（A $）	送回A$中所含的字符个数。
STR$（X）	将X的数值转换成对应的字符串送回。
VAL（A$）	以A$中的第一个非数字字符为界，将其前面的字符转换成数值，送回。
CHR$（X）	送回代码为X的ASCII字符。
ASC（A$）	送回A$中的第一个字符的ASCII代码。
LEFT$（A$，X）	送回A$中的最左边的X个字符。
RIGHT$（A$，X）	送回A$中的最右边的X个字符。
MID$（A$，X，Y）	送回A$中的从字符X开始的Y个字符。
+	用于连接串的运算符。
STORE A	把数值数组A保存到磁带上，不可直接用来存储串数组。
RECALL B	把数组从磁带上装回，B数组必须已经被正确地用DIM定义过。

8. 输入/输出命令

（同时参考LOAD和SAVE，STORE和RECALL）

INPUT A$	在屏幕上显示一个问号（？）等待用户打入一组字符串值给A$。
INPUT "XYZ"；A	在屏幕上打印出XYZ，等待用户打入一个实数值给A。
CETA$	等待用户打入单个字符值给A$，它无需按RETURN键。
DATA X，"Y"，Z	建立READ语句能使用的数据元素表。
READ A$	将下一个DATA的元素赋给A$。
RESTORE	重新从第一个DATA元素开始，用READ来读。
PRINT "X="；X	在屏幕上打印串X=和变量X的值，分号表示紧接着上一项段打印，逗号则表示各打印项之间留有三个制表域（tab field）的长度。 符号？也表示PRINT语句。
IN#6	从#6插口上的外部设备中进行预输入，而不是从键盘（IN#0）上进行。
PR#6	置输出为#6插口上的外部设备，而不是TV屏幕（PR#0）
LET X=Y	把Y的值赋给变量X，LET是选用项。
DEF FNA（X）=X+23／X	定义一个函数A，在后面使用时，A的自变量由定义中的表达式X来代替。FNA（4）将送回值9．75。

9、有关流程控制的命令

GOTO 347	转向347行。
IF X=3 THEN STOP	如果断言X=3为真（非零），则继续执行THEN后的语句。如果断言为假（零）则执行下一行号的语句。

FOR X=1 TO 20 STEP 4…NEXT X

	执行FOR语句和相应的NEXT语句之间的所有语句，首先X=1，然后X=5，X=9等等。直到X>20，才继续执行NEXT后的语句。如果未指明STEP的大小，则表明STEP大小为1。
NEXT X	定义FOR…NEXT循环的底，X是选用项。
GOSUB 33	转向行号为33开始的子程序。
RETURN	标明子程序的结束，返回到最后的GOSUB的下一条语句处执行。
POP	从RETURN地址栈中移出一个地址。
ON X GOTO 397，12，458	转向表中所列出的第X个行号所指出的语句。如果X=2，则转向12行。
ON X GOSUB 397，12，458	转向表中第X个行号的子程序。
ONERR GOTO 4500	在后继语句执行中，如产生错误使得程序转到错误处理程序（在4500行）上执行，而不是显示信息，也不停止程序的运行。
RESUME	在错误处理程序中，使得返回发生错误的语句处执行。

10. 作图和游戏控制

低分辨率图形

GR	置成低分辨率作图方式，把顶端的40×40区域清成黑色，底部留有4行正文显示。
COLOR=X	为画下一个点置颜色（0到15）
PLOT X,Y	把有色的点显示在横坐标为X，纵坐标为Y的位置上，X和Y在0到39范围内，（0,0）是屏幕的左上角。
HLIN XI，X2 AT Y	从点（X1，Y）到点（X2，Y）画一条水

	平线。
VLIN Y1，Y2AT X	从点（X，Y1）到点（X，Y2）画一条垂直线。
SCRN（X，Y）	送回在屏幕上（X，Y）这一点的颜色数。
高分辨图形	
HGR	置高分辨率作图方式（第1页）；把顶端的280×160区域清成黑色，底部留出4行的正文显示。
HGR 2	置第2页的高分辨率作图方式，按整个280×192区域的屏幕清成黑色。
HCOLOR=X	为画下一个点置颜色（0到7）。
HPLOT X，Y	把有色点置在横坐标为X，纵坐标为Y的位置上，X的变化范围从0到279，Y则从0到159（对HGR），或到191（对HGR 2）。（0，0）是左上角。
HPLOT X1，Y1 TO X2，Y2	从点X1，Y1到点X2，Y2画一条线，命令还可以扩充，加上TO Xn，Yn等等点。
SHLOAD	从磁带上装入一个图形表。
DRAW 3AT X,Y	按照先前所装入的图形表中所定义的#3形状来绘图，起始位置为X，Y，所绘的颜色由HCOLOR来确定。
XDRAW 3AT X,Y	按照图形表中所确定的#3形状来绘制，所画的每一点的颜色数是屏幕上那一点原有的颜色数的补数。
ROT=X	为执行DRAW或XDRAW，可将形状进行旋转，ROT=0表示垂直，ROT=16表示顺时针方向右旋90度，ROT=32表示顺时针方向右旋180度等等。
BCLR 2	以颜色数2将整个屏幕涂上颜色。

SCALE=X	置DRAW或XDRAW图形的比例因子（1到255）游戏控制。
PDL（X）	送回游戏控制器X（0到3）所置的值，它可以是在0到255之间进行变化。
PEEK （X-16287）	如果其值>127，表示游戏控制器X（0到2）上的按钮已被按下。
PEEK （—16336）	使APPLE的喇叭发出卡哒声。

11。一些常用数学函数

SIN（X）	送回X弧度的正弦值。
COS（X）	送回弧度X的余弦值。
TAN（X）	送回弧度X的正切值。
ATN（X）	送回X的反正切值，以弧度为单位。
INT（X）	送回小于等于X的最大整数。
RND（1）	每次使用时，总送回一个0到0。999999999的随机数。
RND（0）	再次送回上一次的随机数。
RND（-3）	送回的值为4。48217179E-08，对于每一个不同的负自变量，总是送回相应于不同变量的一个固定值（换言之，一个函数产生的随机数是一定值，与调用次数无关）。使用了这一随机数后，带有正自变量的RND的函数，也产生一固定序列。
SGN（X）	如果X<0，送回值为-1，如果X=0，送回值为0，如果X>0，送回值为1。
ABS（X）	送回X的绝对值。
EXP（X）	送回E的X次方的值。e=2。718289。
LOG（X）	送回X的自然对数的值。
QR（X）	送回X的正平方根。
DEC("FF69")	将十六进制数FF69转换成十进制数。
HEX$（-151）	将十进制数-151转换成十六进制数。

附录H　DOS3.3错误信息表

由于在命令或程序中使用DOS命令不当，会出现如下错误信息，这些错误的代码可由BASIC语句ERR=PEEK(222)得到。

错误代码	错误信息	出错原因
1	LANGUAGE NOT AVAILABLE	没有装入整数BASIC解释程序，而使用了INIT命令。
2．3	RANGE ERROR	命令参数取值超出规定的范围。
4	WRITE PROTECTED	执行写操作时，软盘上贴有写保护。
5	END OF DATA	读操作中，读数据指针越过文本文件的末端。
6	FILE NOT FOUND	文件名拼错，没有找到该文件。
7	VOLUME MISMATCH	DOS命令使用中，卷号不匹配。
8	I/O ERROR	盘片被损坏，或软盘片未经过初始化，或未关闭驱动器的门。
9	DISK FULL	软盘上的文件个数超过额定数，或盘上数据已存满。
10	FILE LOCKED	试图对一个加有封锁的文件进行修改，删除，或重命名操作。
11	SYNTAX ERROR	DOS命令，参数标识符或逗号分隔符用错，或没有启动

		DOS操作系统，而试图使用DOS命令。
12	NOBUFFERS AVAILABLE	同时打开的文件个数超出MAXFILES命令中所规定的文件个数。
13	FILE TYPE MISMATCH	软盘文件的类型与DOS命令所操作的文件类型不匹配。
14	PROGRAM TOO LARGE	没有足够的内存空间来装入新的程序。
15	NOT DIRECT COMMAND	该命令必须在程序中作为语句来使用。

ISBN 7-302-00661-X
TP·227 定价:9.50元

中华学习机CEC-IA用户使用手册

清华大学出版社